

DTIC FILE COPY

2

AD-A228 467

Repository Prototype
System Specification

Document Number CDRL 1590

February 16, 1990

DTIC
ELECTE
NOV 09 1990
S B D

Prepared for:

Electronic Systems Division (PKG-1)
Air Force Systems Command, USAF
Hanscom AFB, MA 01731-5000

Prepared by:

IBM Systems Integration Division
18100 Frederick Pike
Gaithersburg, MD 20879

DISTRIBUTION STATEMENT A

Approved for public release
Distribution Unlimited

REPORT DOCUMENTATION PAGE

Form Approved

OMB No. 0704-0189

Reduce reporting burden for this collection of data by providing estimates of the burden of this collection of data, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of data. Send comments regarding this burden estimate or any other aspect of this collection of data, including suggestions for reducing this burden, to Washington Headquarters Office, Paperwork Project, (0704-0189), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE February 16, 1990		3. REPORT TYPE AND DATES COVERED Final	
4. TITLE AND SUBTITLE Repository Prototype System Specification				5. FUNDING NUMBERS C: F19628-88-D-0032	
6. AUTHOR(S) T. Ward					
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) IBM Federal Sector Division 800 N. Frederick Avenue Gaithersburg, MD 20879				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Electronic Systems Division Air Force Systems Command, USAF Hanscom AFB, MA 01731-5000				10. SPONSORING/MONITORING AGENCY REPORT NUMBER CDRL Sequence No. 1590	
11. SUPPLEMENTARY NOTES					
12a. DISTRIBUTION/AVAILABILITY STATEMENT				12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) A specification of the capabilities, characteristics and tools for a software reuse library. Defines the minimum set of integrated software development and data processing facilities needed to maintain and support a software reuse library.					
14. SUBJECT TERMS STARS, software reuse, software reuse library				15. NUMBER OF PAGES 89	
				16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL		

ABSTRACT

This Technical Report (CDRL Seq. No. 1590) addresses Subtask IR40.2 (Prototype Repository System) of the STARS Delivery Order Task IR40 (Repository Integration). Included in the appendices are the Repository System Specifications developed over the first two Increments (Q and R) of the STARS Competing Primes Contract. These specifications also contain items projected for S-Increment development and evaluation.

The contents of this document contain lessons learned from building and using the current Repository Prototype. Also discussed are aspects of the environment used to support the repository, recommended extensions and modifications to the repository and repository environment, and other accomplishments (as required by DI-S-3591/A).

The contents of the three appendices contain specifications corresponding to the Repository Requirements, the Top Level Design of Repository Prototype capabilities, and the Detail Design of the associated database. These appendices has been formatted for ease of reading and to be reasonably consistent to level of detail and paragraph content as described in the following referenced DIDs:

- DI-MCCR-80025A Software Requirements Specification (A010)
- DI-MCCR-80012 Software Top Level Design (A011)
- DI-MCCR-80031 Detailed Design Document (A012)
- DI-MCCR-80029 Software System Specification (A024)



Accession For	
NTIS GRA&I	<input checked="checked" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By <i>per letter</i>	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
<i>A-1</i>	

TABLE OF CONTENTS

1.0	INTRODUCTION	1
1.1	Identification	1
1.2	Scope	1
1.3	Purpose	1
2.0	REFERENCED DOCUMENTS	2
2.1	Government Documents	2
2.2	Non-Government Documents	3
3.0	TECHNICAL WORK ACCOMPLISHED	4
3.1	Definition of Terms	4
3.1.1	System Concepts	5
3.1.2	Operational Concepts	6
4.0	INFORMATION GAINED FROM THE PERFORMANCE	9
4.1	Repository System Capabilities	9
4.1.1	Components and Component Maintenance	10
4.1.2	Information and Information Exchange	10
4.2	Repository Capabilities	11
4.2.1	Repository System Maintenance and Operation	11
4.2.2	Content Maintenance and Distribution Process	12
4.2.3	Component Supply and Filtering Process	12
4.2.4	Reuse and Development Process	13
4.2.5	Component Part Storage and Distribution	14
4.3	User Interface	14
4.3.1	Screen Layout	14
4.3.2	Action/Hot Keys	15
4.3.3	Search Criteria	16
4.3.4	Help Screens	16
4.3.5	Exiting	16
4.4	Database	17
4.4.1	Commercial Database Tools	17
4.4.2	Table Keys	17
4.4.3	Component Specific Attributes	18
4.4.4	Facet Flexibility	18
4.4.5	Personnel/Organization Representation	18
APPENDIX A.	REPOSITORY SYSTEM REQUIREMENTS SPECIFICATION	20
A.1	Operational Requirements	20
A.2	Repository Management Requirements	21
A.3	User Interface Requirements	22
A.4	Supply Requirements	23
A.5	Search and Access Requirements	24
APPENDIX B.	REPOSITORY TOP-LEVEL DESIGN	26
B.1	Search and Access Overview	26
B.1.1	Screen Format	26
B.1.2	Definition of Terms	27
B.1.3	Methods of Searching	28
B.1.4	User Interface	29
B.2	Search and Access Processing	31
B.2.1	Search Processing	32
B.2.1.1	Component Type Selection Screen	32

B.2.1.2	Attribute Search Selection Screens	33
B.2.1.3	Hierarchical Classification Selection Screens	35
B.2.1.4	Facet Search Selection Screens	36
B.2.1.5	Keyword Search Selection Screen	38
B.2.1.6	STARS Repository Search Criteria Summary	38
B.2.2	Component Processing	39
B.2.2.1	Components returned from search	39
B.2.2.2	Versions of a Selected Component	40
B.2.2.3	Attributes of a Selected Component	41
B.2.2.4	Report Operations Selection Screen	42
B.2.2.5	Problem Reports for Selected Component	43
B.2.2.6	Component Changes to Selected Component Version	43
B.2.2.7	Problem Report Options	44
B.2.2.8	Component Problem Report Submission	45
B.2.2.9	Component Problem Report Submission	46
B.2.2.10	Component Problem Report Viewing	47
B.2.2.11	User Subscription Options	48
B.2.2.12	Subscribed to Items	50
B.2.3	Part Processing	50
B.2.3.1	Parts of Selected Component	51
B.2.3.2	Version of selected part	51
B.2.3.3	Information for Selected Part	52
B.2.3.4	Changes to selected Part Version	53
B.2.3.5	Component Part Copy	54
B.2.3.6	Components using selected component part	54
B.3	Repository Support Overview	55
B.3.1	Component Supply and Filtering	55
B.3.2	STARS Repository Registration Forms	57
B.3.2.1	Component Registration Form	57
B.3.2.2	User Registration Form	59
B.3.2.3	Organization Registration Form	60
B.3.2.4	Contract Registration Form	61
APPENDIX C.	REPOSITORY DETAIL DESIGN DOCUMENT	62
C.1	Data Dictionary	62
C.2	Data Normalization	65
C.2.1	Component Objects	65
C.2.1.1	Definitions	65
C.2.1.2	Relationships	65
C.2.1.3	Object Content	66
C.2.2	Search Objects	67
C.2.2.1	Definitions	67
C.2.2.2	Relationships	68
C.2.2.3	Object Content	68
C.2.3	Component Construction Objects	70
C.2.3.1	Definitions	70
C.2.3.2	Relationships	70
C.2.3.3	Object Content	71
C.2.4	Environment Support / Information Objects	71
C.2.4.1	Definitions	71
C.2.4.2	Relationships	72
C.2.4.3	Object Content	72
C.3	Relational Design	74
C.3.1	Processing Requirements	74
C.3.1.1	Search processes	75
C.3.1.2	Component processes	75
C.3.1.3	Content processes	76

C.3.1.4	Support processes	76
C.3.2	Transformation Rational	76
C.3.3	Process Table Content	77
C.3.4	Support Table Content	79
C.3.5	Additional Indexes	81
C.3.5.1	Component Table	81
C.3.5.2	Component_part Table	81
C.3.5.3	Hierarchy Table	81
C.3.5.4	Component_facet Table	81
C.3.5.5	Problem Table	81
C.3.5.6	Subscription Table	81
C.3.5.7	Component_Change and Part_Change Tables	82

LIST OF ILLUSTRATIONS

Figure 1. Standard Search and Access Screen Layout	26
Figure 2. Examples of Wildcards in Search Fields	29
Figure 3. Overview of the Search and Access Process Flow	30
Figure 4. Example of a Help Window	32
Figure 5. Example of the Quit Window	32
Figure 6. Example of the Component Type Selection Screen	33
Figure 7. Example of the Attribute Selection Criteria Screen	34
Figure 8. Example of the Alphanumeric Search Field Screen	34
Figure 9. Example of the Numeric Search Field Screen	35
Figure 10. Example of the Hierarchical Classification Screen	35
Figure 11. Example of a lower level Classification Screen	36
Figure 12. Example of the Facet Selection Criteria Screen	37
Figure 13. Example of Facet Terms for Selected Facet Screen	37
Figure 14. Example of the Keyword Selection Criteria Summary Screen	38
Figure 15. Example of the Summary Screen Layout	39
Figure 16. Example of the Components Returned From Search Screen	40
Figure 17. Example of Versions of Selected Component Screen	41
Figure 18. Example of the Attributes of Selected Component Screen	42
Figure 19. Example of Change Report Options Screen	42
Figure 20. Example of the Selected Component Problem Report Screen	43
Figure 21. Example of Component Changes to Selected Component Screen	44
Figure 22. Example of Problem Report Options Screen	45
Figure 23. Example of Problem Report Submission Screen	46
Figure 24. Example of Problem Report Answer Screen	47
Figure 25. Example of Problem Report Answer Screen	48
Figure 26. Example of User Subscriptions Options Screen	49
Figure 27. Example of Component Parts for Component Selected for Copy	49
Figure 28. Example of Subscribed to Items Screen	50
Figure 29. Example of Parts of a Selected Component Screen	51
Figure 30. Example of Versions of Selected Component Part Screen	52
Figure 31. Example of the Attributes of Selected Part Screen	53
Figure 32. Example of Component Changes to Selected Part Screen	53
Figure 33. Example of Component Part Copy Screen	54
Figure 34. Example of Components Using a Part Version Screen	55
Figure 35. Example of Dispatching Repository Tools	56
Figure 36. Example of Component Registration Form	58
Figure 37. Example of User Registration Form	59
Figure 38. Example of Organization Registration Form	60
Figure 39. Example of Contract Registration Form	61
Figure 40. Component Object Relationships	66
Figure 41. Search Object Relationships	68
Figure 42. Component Content Relationships	71
Figure 43. Environment Support / Information Object Relationships	72

LIST OF TABLES

Table 1. Repository Data Dictionary 62

Table 2. Component Object Content 66

Table 3. Search Object Content 69

Table 4. Component Content Objects 71

Table 5. Environment Support / Information Objects 73

Table 6. Process SQL table content 78

Table 7. Facet SQL table content 79

Table 8. Support SQL table content 79

1.0 INTRODUCTION

1.1 IDENTIFICATION

This document, Repository Prototype System Specifications (CDRL 1590), contains lessons learned and specifications for an improved repository prototype (Subtask IR40.2 of Delivery Order Task IR40 for IBM Contract Number F19628-88-D-0032).

1.2 SCOPE

Not
This ~~CDRL~~ documents the lessons learned while developing and using the IBM Team STARS Repository System. It also provides rationale for limited changes and specifications for an improved repository prototype based on experience with previous versions of the STARS Repository Prototype. Increment 1 and 2 prototyping efforts have provided a capability to study and evaluate methods for collecting, searching, and accessing reusable information, e.g., software and documents stored in and retrieved from a classical, hierarchical taxonomy based structure. In addition, information stored and retrieved by more user-friendly methods (e.g., sets of related attributes (or facets) defined when reusable component parts are added to the repository) were prototyped and evaluated. Each method was used independently and in combination with other methods. (KR)

1.3 PURPOSE

This document establishes an initial set of specifications for future evolution of the IBM Team STARS Repository System to an improved repository system for the STARS program and eventual release to other DoD or commercial programs. The specifications contained in the appendices are based on the lessons learned and the growing set of information that is contained in the referenced documents and include the following:

- Repository System Requirements Specifications
- Repository Top Level Design
- Repository Database Design

The specifications and designs do not reflect the current state of the IBM Team STARS Repository. The current implementation and capabilities as well as the concepts and guidelines related to the current system are contained in the referenced items, i.e., CDRL 1540, 1570, 1580, and 1600 (Releases A, B, C, and D).

2.0 REFERENCED DOCUMENTS

2.1 GOVERNMENT DOCUMENTS

F19628-88-R-0011	The STARS Competing Primes Lead Contract, Request for Proposal, dated 6 November, 1987.
DI-S-3591	Technical Reports
IBM CDRL 0360	Reusability Guidelines for the STARS Program, dated December 17, 1988
IBM CDRL 0370	Reusable Component Data Analysis for the STARS Program, dated February 10, 1989
IBM CDRL 0380	Consolidated Reusability Guidelines for the STARS Program, dated March 21, 1989
IBM CDRL 0390	Taxonomy Observations for Reusable Ada Components for the STARS Program, dated March 21, 1989
IBM CDRL 0460	Repository Guidelines for the Software Technology for Adaptable, Reliable Systems (STARS) Program, dated February 25, 1988
IBM CDRL 0470	Long-range Repository/Distribution Plan for the STARS Program, dated February 25, 1989
IBM CDRL 0480	Enhanced Repository Formal Demo, dated February 24, 1989
IBM CDRL 0490	Enhanced Repository Demo Report, dated March 3, 1989
IBM CDRL 0500	Specifications for the Prototype Enhanced Repository, dated January 19, 1989
IBM CDRL 0510	Enhanced Repository Implementation Improvement Report, dated March 16, 1989
IBM CDRL 0520	Long Term Configuration Management Plan for the STARS Repository, dated March 17, 1989
IBM CDRL 0530	Implementation of the Prototype Enhanced Repository, dated March 16, 1989
IBM CDRL 0700	Recommended Ada Format Guidelines for the STARS Program, dated October 18, 1988
IBM CDRL 1460	Draft Policies and Procedures for the STARS Repository, dated January 19, 1990
IBM CDRL 1470	Operation and Procedures for the STARS Repository (Draft), dated January 31, 1990 (Final version planned for March 7, 1990).

IBM CDRL 1540	Repository Guidebook (Draft), dated September 14, 1989
IBM CDRL 1550	Repository Guidebook (Draft), dated January 31, 1990
IBM CDRL 1570	Filter Prototype, dated January 24, 1990 (and earlier releases A, B, and C)
IBM CDRL 1580	Taxonomy/Classification, dated January 19, 1990
IBM CDRL 1600	Repository Prototype, Release A, dated July 15, 1989 (and subsequent releases B, C, and D)

2.2 NON-GOVERNMENT DOCUMENTS

ORACLE_SQLREF88 SQL Language Reference Manual, Part No. 778, Version 5.1, September 1988, Copyright(C) 1988, Oracle Corporation, Belmont, California.

3.0 TECHNICAL WORK ACCOMPLISHED

The initial repository prototyping effort was accomplished in the first increment of the STARS Program and the basic capabilities were demonstrated on a workstation, i.e., an IBM PC AT (DOS). Four releases of the second increment prototypes were developed and demonstrated on a large host, i.e., a DEC VAX (VMS). The first increment workstation prototype was rehosted on the larger system with a minimum effort. The only major problem was that the Ada/SQL Bindings for the two systems were different and required changes to the logic that maps the data fields in the database to Ada variables. Minor changes were also required in the STARS virtual interfaces defined to support portability objectives.

The reuse of existing code presented another problem when the reused code required modification and the reused code was the responsibility of other tasks and organizations. Multiple versions had to be maintained until the modified versions were approved and maintained as reusable components. This process resulted in the improvement of reusable components but did result in delays in the delivery of specific repository capabilities.

All search and access capabilities planned for the R-Increment releases (i.e., CDRL 1600A, B, C, and D) of the repository prototype were accomplished including:

- Porting to and integration with new host environment
- First Increment problems and limitations were removed
- Multiple-level Hierarchical Search Criteria was added
- Inclusion of a Catalog generation and browsing feature
- Development of a Component Supply/Filtering mechanism
- Maintenance of user, organization, and contract information
- Component Copy/Component Part Copy with usage statistics
- Component Subscriptions and Report Distribution support
- Integrated Component Problem & Component/Part Change Reports
- Testing of prototype on Virtual Operating System Interface

The implementation and use of the prototype during this increment was used to define the specifications contained in this document. Specific lessons learned will be covered in "Information Gained from the Performance" on page 9 and referenced to specific sections of the specifications. The following paragraphs will define the terms used in this document and describe the major concepts, objectives, and capabilities included in repository prototyping effort.

3.1 DEFINITION OF TERMS

The following terms are used in this document and are consistent with the same terms used in other documents delivered in IQ12 and IR40. The following subparagraphs relate the key terms to either repository system concepts or operational concepts.

3.1.1 SYSTEM CONCEPTS

Repository System

An instance of the software engineering environment, including computer hardware, software tools and environment, standards and procedures, that together provide a complete repository capability.

Repository

An element of the software engineering environment that is used to store reusable items, e.g., software work products, and information describing the items, e.g., the author and owner. The reusable data within the Repository is referred to as **repository contents**.

Library

A collection of reusable components that require special handling, e.g., access control, higher level security, privacy requirements, or limited access during specific software development phases.

Software Work Product/Information

An intermediate or finished work product of software development or maintenance. Software work products include plans, specifications, designs, source and object code, test procedures, reports, demonstrations and other reusable items of information.

Component

A collection of related work products that are intended to be used as a consistent set of information, e.g., Documents, Software Programs, Project Plans, and other types or forms of reusable Information. Regardless of the type or form, a collection of reusable information is always referred to as **Component** in this document.

Component Part

A retrievable element of a component (e.g., requirement and design specifications, software source code or processed forms of that code, a formal document outline, minutes from a project review meeting, etc.). Individual repository operations are performed on these elements, e.g., Browse, Copy, and counting the total number of retrievals. This document refers to these elements as **Parts**.

Variations

Specific occurrences of similar Components or Component Parts that contain equivalent or near equivalent information or function. Each variation may have a separate author, a different owner, different set of data rights, etc. and variations are required to be stored under unique names. Unique names are not required if a unique identifier number is also assigned as defined in the Data Dictionary ("Data Dictionary" on page 62).

Versions

Specific occurrences of Components or Component Parts except that they are incremental updates of the same named object. They may have a different repository entry date, added or corrected information or function, new or modified restrictions to its use, etc. Versions are required to be stored under the same unique name.

Component/Part Owner

The Repository user who had the responsibility to create (or has the responsibility to maintain) a specific Repository (Variation or Version) Component or Component Part, i.e., Repository image of the corresponding reusable item.

Repository System Problem Report

An on-line report generated by any valid STARS Repository user that the user believes applies to the Repository System as a whole. A response report, generated by an authorized user, may be attached.

Repository Component Problem Report

An on-line report generated by any valid STARS Repository user that the user believes applies to a specific Component version. A response report, generated by the Component owner or another authorized user, may be attached.

Repository Component/Part Change Report

An on-line report generated by the owner or another Repository user assigned responsibility for a particular Component version or Component Part Version. The report explains the changes made and should be supplied and processed for acceptance with the new version.

3.1.2 OPERATIONAL CONCEPTS

Component/Part Supply

The process by which the Repository content is established, i.e., the acquisition and installation (for authorized user access).

Component/Part Management

The process by which the Repository content is maintained, tracked, distributed, archived, and deleted. This process is affected by cost of the process and the availability of funds as well as by the individual data rights, risks, and responsibilities assigned to specific individuals or organizations. This could involve the collection and/or distribution of fees associated with the Components or specific Component Parts, i.e., if items associated with fees are to be stored in the STARS Repository.

Repository Supplier

Any individual or organization that supplies data to the Repository, e.g., authors, owners, and distributors of Software Work Products or information related to individual products, e.g., Problem Reports. Included in this definition are individuals or organizations responsible for the testing, verification, validation, certification, maintenance, or any other task where responsibility has been assigned for a task related to Repository content maintenance.

Repository Managers/Librarians/Administrators

Individuals or organizations that are responsible for the operation and maintenance of the Repository System. Included in this definition are individuals or organizations responsible for the testing, verification, validation, certification, maintenance, or any other

defined task, i.e., where responsibility has been explicitly assigned for a task related to Repository System maintenance.

Repository Subscribers/Re-users

Individuals or organizations that use the contents of the Repository including Software Products and information acquired by direct on-line examination, by copy commands, or by other method of retrieval from the Repository.

Repository Gatekeeper/Filtering

The process by which the repository contents are controlled, i.e., admission or removal of Components, Component Parts, and related Reports. This process is based of predefined qualification criteria but can be a manual process and/or by automated tools. The process controls both the quality and integrity of the repository contents, i.e., Components, Components, Component Versions, Component Parts, Component Part Versions, Component or Component Part Change Reports, and Component Problem Reports.

Qualification

The act of establishing that a Component and its Component Parts meet the necessary criteria for admission into the STARS Repository, i.e., making the information available for on-line access and/or distribution.

Disqualification

The act of establishing that a software work product does not meet necessary criteria for storage on the STARS Repository or meets criteria for removal from an accessible state after residing on the Repository.

Migration

The act of moving Components or Component Parts from one library to another or the act of promoting or demoting Components and Component Parts from one state to another within a single library, e.g., from a non-validated state to an independently validated state.

Authorization

The transfer or assignment of rights to individual users for specific repository operations, repository system or content maintenance, or the examination and use of the repository contents or related information.

Distribution

The electronic transfer or shipment of other media containing work products (e.g., Reports, Components or Component Parts) or other types of information from the STARS Repository to authorized users or to the user's organization.

Main Menu Screen

The initial screen presented to users by the STARS Repository System, e.g., the user enters "REPOS" at the VMS command prompt(\$). This screen presents a set of (user) selectable options, e.g., Component Search. Each option initiates specific STARS System processing or lower level menu selection screens.

Component Search Screens

A set of screens associated with Search and Access processing, e.g., a screen to select the search method desired by the user. The screens, or window overlays used in the repository prototype, allow a user to specify search criteria and browse information related to individual components selected by the user's search criteria.

4.0 INFORMATION GAINED FROM THE PERFORMANCE

The following paragraphs describe the lessons learned in the performance of subtasks of Task IR40. The following list summarizes the major lessons learned by major paragraph heading where detail is presented. Some of the items are unique to a repository system where multiple users and terminal are supported or to systems where a simple, rapid response interface is required.

- Repository System Capabilities
 - An immature, untested Repository System will discourage users.
 - System problem and problem correction reports encourage users.
 - Slow or unreliable communication links will discourage users.
 - High-speed links or distributed capabilities encourage users.
 - Forums are good for new users but can be very time consuming.
 - Integrated and tested Repository System tools save user-time.
 - Repository System tools have to be maintained and improved.
- Repository Capabilities
 - Immature, untested Repository components will discourage users.
 - Component problem and problem correction reports encourage users.
 - Validated components and integrated information save user-time.
 - A single classification and search method does not satisfy users.
 - Repository capabilities will have to be maintained and improved.
 - Component standards and supply filtering is a continuous expense.
 - Component storage and distribution methods continue to evolve.
- User Interface
 - A consistent, simple interface is critical and saves user-time.
 - A repository interface must match the user's terminal interface.
 - Screen layouts can impact users not familiar with the standard.
 - Simple commands are better than specific terminal key assignments.
 - Limited training is required for complex information searches.
 - Fast command selection and exit from nested screens are required.
- Database
 - Custom screen generation is faster than with commercial tools.
 - Custom screen generation provides consistent commands and formats.
 - Duplicate component and part names require a unique index field.
 - Search criteria and screen content require multiple table joins.
 - Correct information about users and organizations is important.

4.1 REPOSITORY SYSTEM CAPABILITIES

The concepts and scope established by the Request for Proposal (F19628-88-0011) and direction and concepts defined in subsequent deliverables (e.g., CDRLs 460 and 470) are to be integrated in a Software Engineering Environment (SEE). This includes the possibility of distributing the repository capability to individual DoD and commercial programs as either a stand-alone operation or as a cooperating node in a distributed network of Software Development Environments. Program Management, Configuration Management, and the development and reuse tools require a common user interface and the same database or a standard

interface to multiple databases to be integrated in a SEE. The repository prototyping efforts to date examined the STARS Repository System requirements (CDRLs 0470, 0500, and 0520) as they apply to developing, maintaining, locating, and distributing reusable components.

4.1.1 COMPONENTS AND COMPONENT MAINTENANCE

When a repository is populated with immature components then the users are less likely to select the components for reuse in software involving mission or project critical development efforts. This is offset somewhat if the user knows the components are actively being improved or are also critical to projects responsible for maturing the reusable components. If problem report generation, problem tracking, and component maintenance contracts are in place to correct the problems, then users may choose to start prototyping or implementing with the repository contents. Adequate, reliable problem tracking, problem closure tracking, and Configuration Control (CDRL 0520) capabilities are required to give the users confidence in any repository.

A Problem Reporting capability was developed and integrated with the STARS Repository Prototype to better understand reuse in the context of a total SEE environment ("Problem Report Options" on page 44). This capability allows the user to immediately view all reports relating to the specific version of the component, to generate a new Problem Report, or to generate a response to Problem Reports assigned to the user ("Report Operations Selection Screen" on page 42). The user can also examine Change Reports relating to specific Component versions ("Problem Reports for Selected Component" on page 43) or to specific Part versions ("Changes to selected Part Version" on page 53). If the user first lists all versions of a component ("Versions of a Selected Component" on page 40) or all versions of a part ("Version of selected part" on page 51) then the user is able to examine reports in any version/time sequence.

The integration of reports into the Search and Access process is very efficient from a user's standpoint and provides the information that a user requires to make appropriate initial selections or subsequent replacement of previously selected components.

4.1.2 INFORMATION AND INFORMATION EXCHANGE

Forums and individual-to-individual communication are good in some ways but totally inadequate for very large repositories. Information about all components should not be automatically sent to individual users nor should one expect users to browse forums, no matter how well organized. Forums are good for informal communication but not for timely exchanges.

Users should be requested to Subscribe to specific components that are being used or are being considered for use. Information about components, e.g., problem or change reports, should be linked to the specific library and component part version to which it applies. This approach was prototyped along with the Search and Access capabilities in order that the user

need only receive reports related to the choice of particular components. The capability allows users to browse reports on-line and to subscribe, to un-subscribe, or to view the user's list of current Component subscriptions ("User Subscription Options" on page 48).

4.2 REPOSITORY CAPABILITIES

An objective of the STARS Program was to prototype and evolve a SEE and a repository system to a level where reuse technology could be studied, evolved, and sufficiently matured to move the technology into practice. At the time of the original RFP and as stated in the RFP, it was clear that a single hierarchical classification methods was not sufficient to satisfy all repository requirements. During the first two increments of the STARS Program multiple classification and retrieval schemes have been prototyped and studied. The success of any method or combination of methods depends on the understanding of both the individual doing the classifying and the user involved with the search and access.

A conclusion that can be drawn from user experience with the prototype repository is to let a user select from a set of methods and hide the details of the other methods from that user ("Component Type Selection Screen" on page 32). The repository prototypes (that have been developed and used) merge the search methods into a single screen. This technique is still valid if a user chooses to use a combination of methods in a single search operation, i.e., a single summary screen. The previously entered search criteria is displayed for the user to modify, select, or de-select specific search parameters without re-entering the entire set of search criteria ("STARS Repository Search Criteria Summary" on page 38).

For a very large repository, covering many application domains, Facet and Facet Term lists may get very lengthy. The set of terms and aliases will grow as the repository contents grow to cover other domains; in fact there may come a point where the terminology is not consistent across domains. One solution is to have a hierarchy of search criteria selection, e.g., component type or application. Multiple component types are supported by the Repository Prototype and the set of attributes used to make the selection varies by component type. This technique may prove useful for the other search methods, e.g., facets or hierarchical classifications.

4.2.1 REPOSITORY SYSTEM MAINTENANCE AND OPERATION

It is important to designate a specific budget for the maintenance of the existing capabilities in addition to the budget required to operate the repository on a day-to-day basis. This is particularly important for the first year of operation after the repository is made available to a wider set of users, i.e., the first few months of heavy access will discover errors and performance problems not detected in system test. This budget can cover commercial product upgrades and regression testing. In addition, a budget should be available to automate labor intensive operations that were also not discovered in time to provide automated capabilities,

e.g., in the component supply or distribution process. This budget can cover responding to Repository System Problem Reports and general support to new users of the STARS Repository System.

The prototype repository (CDRL 1600) has several manual operations supported by on-line input forms ("STARS Repository Registration Forms" on page 57) that simplify the maintenance, i.e., the maintainer does not have to know the SQL interface to the commercial RDBMS used to support the repository. Access to the forms are restricted to authorized individuals, i.e., any user should not be able to determine information about any other user.

4.2.2 CONTENT MAINTENANCE AND DISTRIBUTION PROCESS

To support the Component Problem Reports requires the authority to make changes to specific Components and Component Parts. This authority may not be centralized and may require the re-assignment of Problem Reports to other registered users of the STARS Repository (Figure 22 on page 45).

4.2.3 COMPONENT SUPPLY AND FILTERING PROCESS

A major problem encountered with any large repository of information is diverse styles, content, and quality of the information. This will be a significant problem with the STARS Repository when reusable components are collected from various contract deliverables or imported from other repositories. For example, contracts issued by the different Government agencies (USAF, Navy, Army, NASA, FAA, etc.) specify a different set of standards to be used in the generation and delivery of the contract end items (CDRLs). The differences in many cases may not be significant but some preprocessing of the component parts may be required to achieve a level of consistency across the repository contents. This approach is certainly beneficial to potential re-users of the information.

The STARS Repository will require at least a minimum level of consistency of the information provided by the Component Suppliers. This includes electronic data transfer standards for specific media. These standards will evolve with the related technologies and with the set of potential suppliers. Acceptable variations in information format, content, and media should be documented in the STARS Operation and Procedures (see CDRL 1470 for an example) and in User's Guidebooks (CDRL 1550). A single standard that is used by all developers and suppliers will not be achieved in the near future, therefore a filtering and/or correction process is the only way to achieve the desired results in the near term.

For all software components, including the associated documentation, the following automated or manual processing is considered the minimum effort that should be performed on components supplied to the repository. In the case of the compilation and test case execution, it is assumed that the processing is either accomplished by the supplier or the STARS Repository users, i.e., either the repository suppliers or management personnel. The

component parts should include test code or data sufficient to establish the correctness of the reusable source code, even if the component is not a complete compilation units, e.g., code fragment or template.

- Component compilation, execution module builds, and execution
- Computation of various metrics, e.g., complexity, re-usability
- Quality checks, e.g., completeness, style, clarity, correctness
- Security checks, e.g., ownership, copyrights, viruses, etc.

4.2.4 REUSE AND DEVELOPMENT PROCESS

The capabilities of the prototype repository were developed to support reusable components that range in size and scope from fragments to very large reusable systems, e.g., a Software Development Environment (SDE) or an aircraft guidance, navigation, and control system. Reusable components can be either generic or developed for a specific application. The components could have been developed with reuse in mind or be recovered from delivered work products, e.g., technical reports, formal documents, and programming or design language code. Even if the work products were not designed with reuse in mind, there may be small or large portions that are reusable with little work required to extract the information required by the STARS Repository System (see "STARS Repository Registration Forms" on page 57 and CDRL 1460).

To support the reuse of large systems or subsystems, a capability was prototyped to allow the break-up of the large reusable components into smaller reusable components or to allow the systematic build-up of large reusable components from existing reusable components. The smaller reusable components could then be selected individually without retrieving the entire system or subsystem; in fact the capability supports a multiple level break-up and does not require additional component part storage for the levels. The user can then individually select the components desired without having to make the break-up decisions that would otherwise be made by each individual user thus improving overall productivity.

The additional advantage of the component-of-components strategy is that documentation, test drivers, test data, and testing results can be stored with the individual components. Likewise, Problem and Change Reports can be associated with the individual components and not have to be extracted from a collection of reports related to the system or subsystem as a whole. The prototype repository lists any lower level Component as a Component Part with the Component Part type of "COMPONENT". The user can Copy and/or Subscribe to that portion of the reusable system or subsystem that is of interest or immediately reusable.

The separation of components from within other components does require resources in the component supply and maintenance process but these components can be more reliable and traceable, i.e., the user can use the STARS Repository search capability to track down other uses of the component to higher level components to determine just how the lower level component is used. The lower level component can be used in one or more higher level components in the repository, e.g., the STARS Window Manager.

4.2.5 COMPONENT PART STORAGE AND DISTRIBUTION

Component Parts are stored in files external to the relational information required to manage the reusable part. In order to retrieve these files for distribution to repository users, there needs to be a standard naming convention established. The ideal convention would store a platform-independent name in the database so that this name will not need to be changed when moving or installing the repository onto another platform.

The system developed for the prototyped repository stores all file names as integers and a mapping algorithm converts this number into a pathname and filename. It is recommended that this approach be kept; the use of a different algorithm is supported, though, by the separation of all mapping functions into one Ada package. This package may be changed in any way (except for the parameters) without affecting the program, e.g., the pathname and filename algorithm can be changed to be compatible with the STARS Virtual Operating System Interface (VOSI).

4.3 USER INTERFACE

This section reviews the lessons learned pertaining to the means of relaying information to the user. The following topics will be discussed:

- Screen Layout
- Action/Hot Keys
- Search Criteria
- Help Screens
- Exiting

The prototype repository (CDRLs 0530 and 1600) used a form of display screen management that allowed the user to maintain the context of the search and access process, i.e., the STARS Window Manager. A window placement algorithm positioned the next window as close as practical to the current position of the cursor. This algorithm also sized the window to match the data to be displayed without the user having to scroll the display, unless sufficient space was not available for an entire list to be displayed. This proved useful for complex search and browse sessions (as possible in Figure 3 on page 30) but was confusing to new or casual users of the repository.

4.3.1 SCREEN LAYOUT

The screen layout as implemented in the R-Increment has been found to be easy to use and interpret. This window layout contains a list of available actions on the top line, followed by the title of the window on the second line, an optional subtitle or column heading on the third line, an information line on the bottom of the window, and all data on the remaining (scrollable) lines. The user may choose which object in the window to perform an operation on by moving the highlighted selection bar

to that object, and then pressing the associated action key, as described in "Search and Access Processing" on page 31.

This window approach presents a straight-forward, context sensitive interface and is similar to that used in many commercial products from a wide range of vendors. It should thus be familiar to a wide range of users, decreasing the time needed by the user to learn the interface to the STARS Repository System.

Specifications in the appendices do not require specific window or screen layouts, i.e., the implementation can be complete overlays of the previous screen, windows with one or more screen layout algorithms, or other forms of split-screen display that contain the required content and format (Figure 1 on page 26). Highlighting has been included in the prototyping efforts with color emphasis on help (Figure 4 on page 32) and warning (Figure 5 on page 32) or error messages as well as instructional messages from the search and access process. This has proven useful, particularly with the multiple window overlays that appear to clutter the display screen.

4.3.2 ACTION/HOT KEYS

The actions and operations listed in the action bar were originally provided by assigning a particular operation to a specific function key. It was soon realized, however, that not all platforms provide these function keys. An alternative was derived that maps an operation to a standard character on the keyboard. For example, the function that processes the selection criteria and returns the components matching that criteria is represented in the action bar as 'Process' and is performed by pressing the 'P' key on the keyboard. This mapping is conveyed to the user by highlighting the character in the action bar, e.g., 'P' is highlighted and capitalized in 'Process'.

Action (or Hot) Keys were added and proved easy to use since the alphanumeric keyboard was not used except to enter numeric or alphanumeric search parameters, i.e., fields in the standard user input forms ("Attribute Search Selection Screens" on page 33). Simple switching in or out of "input" mode allowed the standard keyboard to be used for all Action Keys. If PFKs are used they should not be described on the Command Bar thus allowing the keys to be remapped to be consistent with the application programs or tools used with the STARS Repository .

The use of these Action Keys is dependent upon each function being mapped to a unique key for each window. For example, it occurs that some windows provide both an EXPLAIN and EXIT function. In order to prevent a collision of function-to-key mapping, both of these functions cannot be mapped to the 'E' key. A reasonable solution is to assign EXPLAIN to the 'E' key, and EXIT to the 'X' key so that the action bar would look like

Explain . . . eXit

4.3.3 SEARCH CRITERIA

The search criteria as implemented in the R-Increment is displayed in alphabetical order on one screen. This appears somewhat overwhelming to the novice user. An alternative approach has been derived that first presents only the types of components available on the screen, and then, once this value has been assigned, the other attributes may be manipulated by selecting the attribute type from the action bar. A window is also provided that lists all attributes that have been selected as search criteria by the user and the values assigned by the user to these attributes. See Figure 15 on page 39 for a more detailed explanation of the screen layout.

The primary advantage to this approach is that the user is not inundated with information the first time they access the Repository. The attributes are still there for the user, but they are arranged in an ordered way so particular users can apply the search mechanism that is best for the user or application. Other search methods can then be ignored. For example, for a person who prefers to use a simple keyword search, a keyword mechanism should be provided. That user may base his search purely upon this means, or may combine it with any other search means, such as the hierarchical taxonomy or faceted taxonomy.

4.3.4 HELP SCREENS

It has been found that the greatest assistance that can be provided to a user is a help screen for each window that describes what the window does, the functions allowed, and the means of activating those functions. This method of conveying information was selected rather than putting the information directly on the screen for several reasons. First, a separate help window allows more information to be displayed, especially if the help window may be scrolled and/or paged. Second, this information will not need to be seen by users as they become more familiar with the system. Placing this information in a separately called help window allows it to be seen by the user who needs it, but does not force itself on the more experienced user. Third, more screen space is available for data, pertaining to the process selected, instead of being used to display static help information. For example, the functions provided can be shortened to one word, or abbreviation of the word, in the action bar and a more detailed explanation of the function can be given in the help window.

All of the help information has been collected into a single Ada package specification so that this information may be customized to each specific platform.

4.3.5 EXITING

The original Repository only allowed the user to exit from one window to the previous window. It became evident, however, that a fast exit was also required to allow the user to exit the system completely, regardless

of where they were. This was especially true when the user was several screens or window-overlays into the search and browse process. Instead of requiring the user to repeatedly press the exit key many times, a Quit key was provided, which, after verifying that the key was not accidentally pushed, would return the user to the screen immediately preceding the opening screen of the Repository search process, i.e., to the STARS Repository System Main Menu.

4.4 DATABASE

This section describes the lessons learned pertaining to the database. The tables are discussed in more detail in "Appendix C. Repository Detail Design Document" on page 62. The following topics will be discussed:

- Commercial Database Tools
- Table keys
- Component Specific Attributes
- Facet Flexibility
- Personnel/Organization Representation

4.4.1 COMMERCIAL DATABASE TOOLS

A commercial Relational Database System was used to support rapid prototyping in Task IR40 (CDRL 1600). In some cases the prototyped capability was generated by the report generation capability of the commercial RDBMS, i.e., for Change and Problem/Response Reports and for User and Organization Registration Forms. This was productive but limits the portability of the prototype to other commercial RDBMS.

Search and Access capabilities are implemented with the Ada/SQL binding supported by the commercial RDBMS (CDRL 1600). The Ada/SQL Bindings being developed for the STARS Program should be used to replace the commercial bindings. The earlier prototype (CDRL 0530) used an early STARS Program developed Ada/SQL Binding but the implementation proved unsatisfactory (CDRL 0510). Also, the forms should be re-implemented to be independent of the commercial RDBMS. In a few cases this has been accomplished (CDRL 1600) and the performance improved and the user interface, i.e., screen format and user controls, was then consistent with other Search and Access screens.

4.4.2 TABLE KEYS

The components and parts were originally identified in the database by the unique combination of name, type, and version. This scheme did not allow different components and parts of the same type to have the same name, which does occur in real world situations. The alternative approach recommended for the next improvement of the Repository Prototype assigns a unique random number to each component and part. In addition, since

versions of components and parts are considered separate entities, a unique random number is also assigned to each version of a component and part. In the same way, a unique random number is assigned to reports (both change and problem reports) contracts, organizations, and organizations.

4.4.3 COMPONENT SPECIFIC ATTRIBUTES

As more than one type of component was supported, it became apparent that most attributes are not applicable to all component types. In fact, a separate classification scheme evolved for each type of component supported. In order, then, to keep track of which attributes applied to which components, the attributes which were common to all components were kept in the COMPONENT table and all others were removed to separate tables, one table for each type supported. Although this means another table join is needed to use this scheme, it saves space in the COMPONENT table and presents a better abstraction of the component within the database. Furthermore, the addition and removal of these type-specific attributes affects only those components of the one type, not all components.

4.4.4 FACET FLEXIBILITY

Analogous to the problem mentioned in the above section, it has been found that no one facet taxonomy will accurately describe the domain of all component types. It is therefore necessary to have a different taxonomy for each component type. To provide multiple taxonomies, the facets and facet terms are stored in a separate table and linked to the COMPONENT table by the unique component identification number. An added benefit of this arrangement is that a component may have more than one facet term for a specific facet. This benefit can best be seen in cases where a software component abstracts a data object. In such a case, many facet terms are needed for the FUNCTION facet to describe all the operations provided by the component.

4.4.5 PERSONNEL/ORGANIZATION REPRESENTATION

It was found that people, more often than not, fill multiple roles (Librarian, Re-user, Supplier, Evaluator, etc.) in the operation and use of the STARS Repository System. To support this observation, a single table was designed to hold the vital information, such as name and username, for all users, i.e., the people associated with an organizations. Information describing organization and contracts related to the repository contents are maintained as separate tables. This eliminates redundancy and makes it simpler to maintain accurate, up-to-date information required to manage the repository. This same strategy was also used for other common information that was to be used for multiple Components and Component Parts, e.g., Nation table that contains the name and status of

nations other than the United States with respect to component supply and distribution (Figure 43 on page 72).

It was also realized that mailing addresses of people are not necessarily static objects, so the mailing address is only kept for the organization. This scheme makes the assumption, which is considered valid for all cases, that an organization will know how to reach the people associated with it. The information was also used in the prototype repository to support particular capabilities, e.g., to ensure accurate information when system or component error report are generated or system and response messages are routed.

APPENDIX A. REPOSITORY SYSTEM REQUIREMENTS SPECIFICATION

The STARS Repository System shall have the following tools, capabilities, and characteristics. The list is included to define the minimum set of integrated software development and data processing facilities required to maintain and support the STARS Repository System and the repository contents.

A.1 OPERATIONAL REQUIREMENTS

The STARS Repository System shall support the following capabilities and performance requirements:

1. A system with dial-in communications for remote access and capable of 24-hour operation and the distribution of information in both hard-copy and electronic form. The information may be mail or user notes and reports among registered, authorized users. This information shall include components and component parts selected from the STARS Repository by a registered, authorized user. Authorized users include Suppliers, Managers, and Subscribers (i.e., re-users of the Work Products or information).
2. System and database performance sufficient to ensure two(2) second response for trivial transactions, e.g., changing the selection criteria from a single screen, and five(5) seconds on a trivial search operation, e.g., returning a single screen of components meeting the pre-specified selection criteria. These times are to be met at least 97% of the time for TBD simultaneous, remote users.
3. System and database capacity sufficient to retain all Operational Procedures, Standards and Guidelines, tools for component collection, maintenance, search, access, and distribution of selected components, and storage of TBD million lines of Ada code and associated documents.
4. One or more Ada development environments for individual users to analyze, compile, build, and test selected components. Each supported environment shall include at least one validated Ada compiler, text and Ada language-sensitive editor, an Ada run-time environment, and a user-friendly Ada program debug capability.
5. One or more Relational Database Management Systems (RDBMS) to store, control, access, and manage the repository contents. This includes report generation and other tools to monitor content usage and to control the set of authorized users and organizations. Generated reports include catalogue, list of components retrieved, list of parts retrieved by component, list of subscriptions by organization, list of users by organization, and list of components by contract or supplier.
6. Tools for the preparation, formatting, and printing of documents in the Standardized Generalized Markup Language (SGML).

7. A configuration management system to control information added to, maintained in, or deleted from the repository. This includes repository content and quality standards as well as reports generated with respect to the repository system or to selectable components in the repository.
8. An access security system that provides for the registration of users and the organizations they represent or support. This access security includes password protection mechanisms, see CSC-002-85 DoD Password Management Guideline, April 12, 1985 (referenced in F19628-88-R-0011).
9. Other applicable requirements as defined in F19628-88-R-0011 under Software Engineering Environment (Preliminary System Description), under STARS Repository System (Preliminary System Description), and within referenced documents (Applicable Documents List).

A.2 REPOSITORY MANAGEMENT REQUIREMENTS

The STARS Repository or Repository System shall support the following management and operational requirements:

1. Management of individual files containing reusable information or selectable elements of that information. Information, information files, and related reports shall be maintained in separate Libraries or other form of partitioning as required to meet access security or other control requirements and restrictions with respect to access, distribution, and repository storage hierarchies, e.g., archives. These managed, selectable elements will be called **parts**.
2. Capability to define and redefine Repository Libraries, to limit access to specific libraries by user and by organization, to support library selection by authorized users for a single search operation, and to send/receive an entire library to/from a remote, compatible repository.
3. Classification of parts according to general content. General content will be conveyed by the **part type**.
4. Capability to define or redefine the set of part types.
5. Management of multiple versions of a part.
6. Designation of a single "default" version of a part which is defined to be the best part version to use. The determination of what part is "best" is left to the part supplier or a part (quality) promotion process.
7. Capability to group parts of differing types (of information) into a cohesive collection. These collections will be called **components** and each unique mapping of part types to components will be called **component type**.
8. Capability to define or redefine the set of component types.

9. Management of multiple versions of a component
10. Designation of a single "default" version of a component which is defined to be the best component version to use. The determination of what component version is "best" is left to the component supplier or a component (quality) promotion process.
11. Parts must be shareable among components and the users provided a method to determine which component version is using a particular part version.
12. Components must be definable as a set of parts where one or more of the parts may be an independent, selectable component, i.e., components may have unique parts but contain a multi-level set of components. Each level of components is indicated by a part type of **component**. The fact that a component may be logically in one or more components shall have no impact on individual subscriptions or copy operations.
13. Report writing, storage and retrieval mechanism shall be available for access by users. The reports shall support problem and problem resolution documentation with respect to the Repository System or specific versions of a component. This shall include the documentation of specific changes that have been made in the creation of new component or component part versions. Problem reports shall have a display and control status of at least "open", "answered", "approved" and "closed" where the answering, approving, or closing is by authorized users only.
14. Tools shall be available to identify and maintain a list of valid users of the system as well as other ancillary data required by law or the contract to operate the STARS Repository System (CDRL 1460). The user information includes their mailing address, the organization they represent or support, and an optional network address. The organization information can provide the users address, the organization's address, and an optional network address.
15. Tools shall also be available to manage the components and component parts in the repository as well as associated reports, e.g., catalogs, problem reports, and change reports. The tools should also provide for automated collecting, maintaining, distributing, and retiring components, components parts, and (repository or system) reports. The explicit requirements for collecting, filtering, and loading information into the repository are covered under Supply Requirements below.

A.3 USER INTERFACE REQUIREMENTS

The STARS Repository and Repository System shall be designed and implemented to meet the following requirements:

1. The user shall be notified at sign-on, i.e., when the Logon Password has been validated, how to start the STARS Repository System processing. The user shall also be notified of any current or planned

conditions that could affect the user or the level of service from the repository.

2. The Component Search and Access process shall be implemented to be initiated from the STARS Repository Main Menu. This menu shall also contain entries to initiate or review Repository System Problem Reports as well as to access general information describing all capabilities that are available to the user. The Main Menu or lower level menus are to be provided for authorized users to initiate all processing related to repository component supply (e.g., filtering), maintenance (e.g., migration), and reuse (e.g., search, access, and distribution).
3. All interactive interfaces, i.e., user keyboard inputs and display screen outputs, shall have a standard layout or format and terminology that is consistent for all repository commands, process responses, and other information presented by the Repository System. Included in the latter are information presented in all user error response, user help, or other forms of information presented to the user.
4. The system shall provide error messages for all invalid user requests or commands and confirmation messages for all commands that result in database changes, extended processing that occur before responses, and on any command that may result in loss of information for the system or user. The latter shall also be in the form of a user confirmation input if the processing involves major impact to the database, e.g., deletion of a report, or may lead to significant loss of time or information to the user, e.g., to quit the search process at any processing point other than the first screen below the Main Menu.

A.4 SUPPLY REQUIREMENTS

The component supply process shall be automated to the greatest degree practical to reduce the cost of the repository operation and to increase the quality of the repository content, i.e., Components, Component Parts, and related Reports. The STARS Repository shall contain the following tools and capabilities to ensure a predefined level of quality is met by all components before they are added to the repository (CDRL 1460).

1. Ada code compiler, executable module generator, and an execution environment sufficient to test components that meet a specific set of predefined STARS guidelines and standards shall be available in the STARS Repository for authorized users.
2. A capability to initiate existing and future Software Engineering Environment (SEE) processing including STARS Program specific automatic component part analysis tools (e.g., spelling checkers, complexity and other metric summaries, control and data flow analyzers, security and completeness testing) shall be provided.
3. A capability to sequence the execution of a set of processes or analysis tools and capture the tool outputs, e.g., analysis results, for storage as a Component Part or Report shall be provided.

4. Completeness (e.g., statement data rights, abstracts, keywords, and other prologue information) and correctness checks (e.g., spelling, syntax, and semantics) shall include all items listed in the STARS Repository Guidebook.
5. The above capabilities shall be designed and implemented to be re-used or ported to other SEEs, e.g., the various component supplier or re-user environments. This shall not preclude the use of commercial products in the STARS Repository that require purchasing of license by the supplier for the supplier's development environment. The capability shall allow a supplier to substitute comparable tools (e.g., spelling checkers and compilers) but does not imply that such a capability is imposed on the suppliers.

A.5 SEARCH AND ACCESS REQUIREMENTS

The STARS Repository shall be designed and implemented to meet the following requirements:

1. All search processing shall be performed at the component level.
2. Support a component attribute storage mechanism as well as alphanumeric and numeric search attribute.
3. Support a hierarchical classification mechanism to catalog components.
4. Support a faceted classification mechanism to catalog components.
5. Support a keyword storage mechanism as well as a user selected keyword searching mechanism.
6. Support the above searching mechanisms for a single search.
7. Provide a catalog generation and browsing capability.
8. Provide hard-copy generation and distribution of all repository procedures, required forms, and catalog to potential users of the repository.
9. List all versions of a specified component.
10. List all versions of a specified part version.
11. List all part versions of a specified component version.
12. List all components which use a specified part.
13. Browse individual part versions.
14. Generate or answer problem reports for a specified component version.
15. List and allow browsing of problem and change reports for a specified component version or change reports associated with a specified part.

16. List all reports by the date created with the latest first.
17. Copy to the default directory of the current user.
18. Copy all required parts or the set of user specified parts of a specific component version using a default naming convention.
19. Copy individual part versions to the file name and directory path provided by the user, i.e., within the default directory assigned to the user.

The requirements section of this document provides additional detail and rationale for this screen layout.

B.1.2 DEFINITION OF TERMS

Attribute	specify the use of numeric and alphanumeric attributes in the search criteria
Browse	displays component, part, or object highlighted by the cursor in browse mode
Changes	displays the list of change reports associated with the selected component or component part version
Class	specify the use of a hierarchical classification in the search criteria
Command Bar	the top bar of the STARS repository search screens that displays the action choices available to the user with the currently active screen. The actual choices will vary from screen to screen. Each of the actions is associated with a unique 'hot key', that is displayed in upper case and highlighted (i.e., the 'X' in eXit, the 'E' in Explain, the 'S' in Specify, etc.). Therefore, the action can be initiated via its designated 'hot key'.
Copy	initiates the copy process for the user selected component or individual component part
Drop	allows the user to drop a subscription to the component highlighted by the cursor
Exit	returns the user to the previous screen
Explain	displays a description of the current, highlighted selection choice (the action verb 'Explain' is synonymous with the term describe)
Facet	specify to use of facets in the search criteria
Help	a screen explaining the current screen being displayed, available user actions, and program function key mappings
Info	provides the user with attribute information about the object highlighted by the cursor
Keyword	specify the use of keywords in the search criteria
Next	advances the user to the next level of classification, when applicable
Parts	displays the list of parts that make up the component highlighted by the cursor. Some component parts are op-

tional and others are required for the component to be considered 'complete'. An asterisk will be displayed to the left of the required parts.

Process	executes a repository query based on the search criteria specified by the user (the action verb 'Process' is synonymous with the term search)
Quit	returns the user to the STARS Repository Main Menu
Reports	allows the user to access the change and problem reports associated with the object highlighted by the cursor
Reset	resets (clears) all search parameter selections indicated on the current screen
Specify	allows assignment or reassignment of values to attributes.
Select	allows the user to select the object highlighted by the cursor; usually the enter key will perform this action
Subscribe	allows the user to subscribe to the object highlighted by the cursor
Summary	displays the user's current search criteria. Criteria on the screen highlighted by the cursor can be toggled off and on.
Top	returns the user to the search filed screen after progressing several layers into the component search hierarchy
Versions	displays on-line component version data for the object highlighted by the cursor

B.1.3 METHODS OF SEARCHING

The component searching methods provided in the STARS Repository shall be:

- faceted classification scheme
- hierarchical classification scheme
- specification of search attributes
- specification of search keywords

The faceted classification scheme shall allow the specification of a name that represents a perspective, viewpoint, or dimension of a particular domain. Each facet shall be represented as defined in (CDRL 1590) as having one or more facet terms and an alias list associated with it.

The hierarchical classification scheme shall allow specification of a subtree of the hierarchy as well as individual, terminal, or leaf-node hierarchical entries as search criteria.

Attribute search criteria shall typically be defined as an exact value or as a range of numeric or alphanumeric values. Alphanumeric attributes can also be specified as a list of values to search or a pattern using wildcards.

The Keyword search shall provide the user with the ability to indicate keywords as their search criteria.

If the user elects to use the attribute or keyword option as his search criteria he may expand the attribute or keyword specified through the use of a 'wildcard' parameter.

The characters selected for the 'wildcard' parameters are the (percent symbol) "%" and the (underscore) "_" (ORACLE_SQLREF88, page 1-31). The percent sign is used to indicate that one or more characters may be placed in the space, whereas the underscore indicates that only a single character may fill the space, as shown in Figure 2.

Query	Possible Results
JO%	JOHN, JOHNSON, JOB, JON
JO%N	JOHN, JOHNSON, JON
JO_	JOB, JON
JO_N	JOHN
12-MAY-%	12-MAY-1988, 12-MAY-1989
_ -MAY-1989	01-MAY-1989 ... 31-MAY-1989

Figure 2. Examples of Wildcards in Search Fields

B.1.4 USER INTERFACE

As stated in "Appendix B. REPOSITORY TOP-LEVEL DESIGN" on page 26 the component search process of the STARS Repository System is available to the user through a set of search support screens. Selecting the 'Component Search' option on the STARS Repository System Main Menu provides the user interface to the Component Type Selection Screen. An example of the component type selection screen is shown in "Component Type Selection Screen" on page 32.

Figure 3 on page 30 provides an overview of the screen support that shall be provided for the component search process.

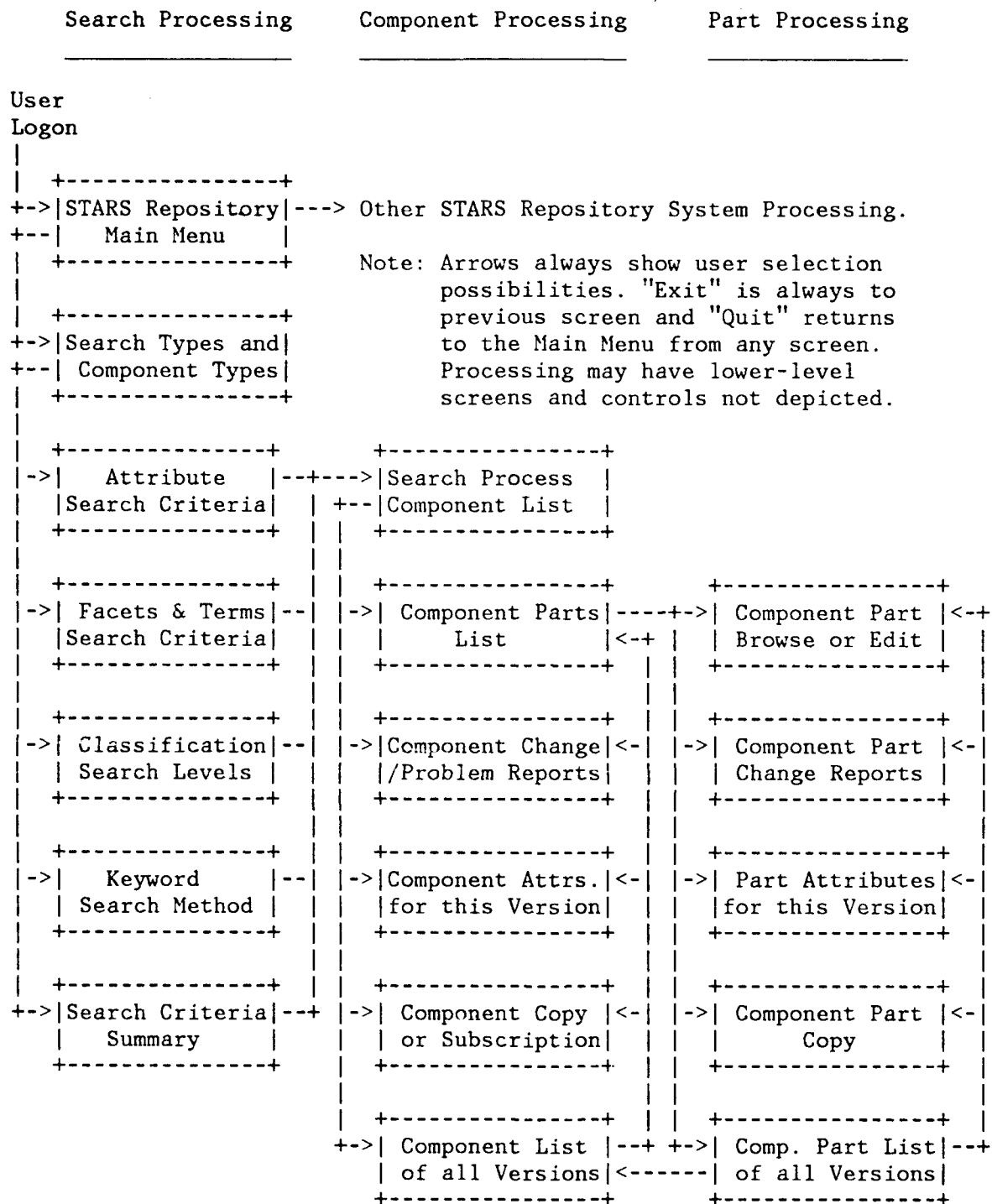


Figure 3. Overview of the Search and Access Process Flow

The following is a list of tentative titles for the component search screens.

- Classifications for the selected component type
- Components returned from search
- Attributes of selected component
- Pattern for a selected search field
- Terms for the selected Facet
- Parts of selected component

- Change Report Options
- Problem Reports for Selected Component
- Information for Part
- Versions of Selected Component
- Versions of Selected Component Part
- Components using Selected Component Part
- Component Changes to Selected Component
- Changes to Selected Component Part version
- User Subscription Options
- Subscribed to Items

B.2 SEARCH AND ACCESS PROCESSING

A set of descriptions of the component search and access screens that were referenced in "User Interface" on page 29 are presented in the following paragraphs. These descriptions will:

- state the primary purpose of the screen
- provide an example of the screen layout
- indicate the valid actions available on each screen.

The following operations shall apply to all of the component search and access screens.

1. The cursor will be positioned on the object to be processed (in the selection list area of the screen) by the user.
2. Pressing the 'hot key' associated with an action (displayed in the action bar) will apply the action to the highlighted object. The 'hot key' is the letter in the action verb that is highlighted and in upper case.
3. Pressing the Help action key will present a window to the user containing additional information about the purpose of the screen, the functions available, and the means of performing those functions. An example help window is shown in Figure 4 on page 32.

```

+-----+
| This window presents the component types |
| in a selection list format.  Once one of |
| the types is selected, the attribute types |
| on the action bar may be chosen for fur- |
| ther specification.  The following keys |
| are available:                          |
| A      - Specify a character or numeric |
|         attribute                      |
| C      - Specify the hierarchy value    |
| E      - Display an explanation of the |
|         highlighted object              |
| F      - Specify a facet term           |
| H      - This window                    |
| K      - Specify a keyword attribute    |
| Q      - Return to the STARS Repository |
|         Main Menu                      |
| S      - View or specify all the attributes |
|         related to the selected type    |
| X      - Return to the previous screen  |
| Enter - Select the highlighted object   |
+-----+

```

Figure 4. Example of a Help Window

4. If the Quit action key is pressed, a screen shown in Figure 5 will ask the user to enter 'Y' to exit to the STARS Repository Main Menu. If any other key is pressed, the user will be returned to the previous screen.

```

+-----+
| Do you wish to exit the Component Search tool? |
| Press 'Y' to exit, any other key to cancel.    |
+-----+

```

Figure 5. Example of the Quit Window

B.2.1 SEARCH_PROCESSING

This section will describe the windows used to establish the search criteria needed to find a specific set of components. The first section, "Component Type Selection Screen", presents the first screen. From this screen, the other attribute specification screens may be invoked.

B.2.1.1 Component Type Selection Screen

The purpose of this screen shall be to accommodate:

- selection of a component type
- specification of a search method.

The component types appear in the selection list portion of the screen (i.e., DOCUMENT, SOFTWARE, etc.). An example of the component type selection screen is shown in Figure 6 on page 33.

Attribute	Class	Explain	Facet	Help	Keyword	Quit	Summary	eXit
STARS Component Type Selection								
DOCUMENT								
SOFTWARE								
...								
Select the component type before initiating the specific search type.								

Figure 6. Example of the Component Type Selection Screen

Valid actions on the screen are

Attribute	specify the use of alphanumeric or numeric attributes in the search criteria
Class	specify the use of a hierarchical classification in the search criteria
Exit	Returns the user to the previous screen. In this case, performs the same function as Quit
Explain	displays a description of the current, highlighted selection choice
Facet	specify the use of selected facets in the search criteria
Help	a screen is displayed to explain the purpose of the current screen and information about what actions are valid and how they are invoked
Keyword	specify the use of selected keywords in the search criteria
Quit	returns the user to the STARS Repository Main Menu
Summary	displays the user's current search criteria. Objects highlighted by the cursor can be toggled off and on.

B.2.1.2 Attribute Search Selection Screens

The purpose of this screen shall be to show the current selection criteria for the Attribute selection method. An example of the Search Criteria Summary (Attribute Selection Method) screen is shown in Figure 7 on page 34.

	Explain	Help	Process	Quit	Reset	Specify	eXit

STARS Repository Attributes for SOFTWARE							
-----+-----							
Sel	SEARCH FIELDS (Current Search Value)						
-----+-----							
*	AUTHOR (Like GRADY%)						
*	DATE_ENTERED (Like __-DEC-1989)						
	DEFAULT_VERSION						
	NAME						
	SIZE						
...	...						
-----+-----							
Select/deselect the attributes before processing							

Figure 7. Example of the Attribute Selection Criteria Screen

Valid actions on the screen are

Exit	returns the user to the previous screen
Explain	displays a description of the current, highlighted selection choice
Help	a screen is displayed to explain the purpose of the current screen and information about what actions are valid and how they are invoked
Process	executes a repository query based on the search criteria specified by the user
Reset	resets (clears) all search parameter selections indicated on the current screen
Quit	returns the user to the STARS Repository Main Menu
Specify	allows assignment or reassignment of values to attributes
Select	toggle individual object assignments by highlighting the object, which is accomplished by pressing the enter key

The screen shown in Figure 8 shall allow the user to indicate the pattern to be used when searching a particular alphanumeric search field.

	Help	Process	Quit	Specify	eXit
Pattern for: NAME					
Pattern: _____					
Press [Enter] with no value to abort the Attribute specification.					

Figure 8. Example of the Alphanumeric Search Field Screen

The screen shown in Figure 9 on page 35 shall allow the user to indicate the pattern to be used when searching a particular numeric search field.

Help	Process	Quit	Specify	eXit
Numeric Range for: SIZE				
Lower bound: _____ (minimum value - 100)				
Upper bound: _____ (maximum value - 10000)				
Press [Enter] with no value to abort the Attribute specification.				

Figure 9. Example of the Numeric Search Field Screen

Valid actions on the screens are

Help	a screen is displayed to explain the purpose of the current screen and information about what actions are valid and how they are invoked
Process	executes a repository query based on the search criteria specified by the user
Quit	returns the user to the STARS Repository Main Menu
Specify	allows assignment or reassignment of values to attributes
Exit	returns the user to the previous screen

B.2.1.3 Hierarchical Classification Selection Screens

The purpose of this screen shall be to assist in the tailoring of the search criteria through a more 'complete' component classification. An example of the Classifications for the selected component type screen is shown in Figure 10.

+	Help	Next	Process	Quit	Specify	Top	eXit

STARS Repository Classification							

CLASSIFICATION							

0	STARS PRIMES - INTERCHANGE						
1	STARS Repository Taxonomy						
2	BOEING DELIVERABLES						
3	IBM DELIVERABLES						
4	UNISYS DELIVERABLES						
	...						

Select/deselect the level before processing.							

Figure 10. Example of the Hierarchical Classification Screen

Valid actions on the 'classifications' screen are

Help	a screen explaining the current screen being displayed, available user actions, and program function key mappings
-------------	-------------------------------------------------------------------------------------------------------------------

Exit returns the user to the previous screen

Next advances the user to the next level of classification, when applicable.

Process executes a repository query based on the selected search criteria specified by the user (see "Components returned from search" on page 39)

Quit returns the user to the STARS Repository Main Menu

Specify selects the highlighted object as the search field and returns to the previous screen. May be performed by pressing enter.

Top returns the user to the search filed screen. See "Hierarchical Classification Selection Screens" on page 35

The following is an example of how the 'next level' classification screen will be formatted assuming the STARS Repository Taxonomy was selected.

Help	Next	Process	Quit	Specify	Top	eXit
STARS Repository Classification						
CLASSIFICATION						
1.1 Software Building Blocks						
1.2 Software Development Support						
1.3 Application Support						
...						
Select/deselect the level before processing.						

Figure 11. Example of a lower level Classification Screen

B.2.1.4 Facet Search Selection Screens

The purpose of this screen shall be to show the current selection criteria for the Facet selection method. An example of the Search Criteria Summary (Facet Selection Method) screen is shown in Figure 12 on page 37

	Explain	Help	Process	Quit	Reset	Specify	eXit

STARS Repository Facets for SOFTWARE							

Sel	FACET (Current Facet Term)						

*	FUNCTION (Eq ANALYZE)						
	LANGUAGE						
	MEDIUM						
*	OBJECT (Eq STATISTICS, REPORTS)						
...	...						

Select/deselect one or more facets before search processing.							

Figure 12. Example of the Facet Selection Criteria Screen

Valid actions on the screen are

Exit	returns the user to the previous screen
Explain	displays a description of the current, highlighted facet
Help	a screen is displayed to explain the purpose of the current screen and information about what actions are valid and how they are invoked
Process	executes a repository query based on the search criteria specified by the user
Quit	returns the user to the STARS Repository Main Menu
Reset	Resets (clears) all the selected facets and their facet terms
Select	toggle selection of individual object assignments by pressing the enter key
Specify	allows assignment or reassignment of values to attributes.

The screen shown in Figure 13 shows the facet terms associated with the facet and search field.

+-----+-----+-----+-----+-----+-----+											
Explain		Help		Process		Quit		Reset		eXit	
+-----+-----+-----+-----+-----+-----+											
Terms for FACET: FUNCTION											
+-----+-----+-----+-----+-----+-----+											
Sel		FACET TERM				ALIASES					
+-----+-----+-----+-----+-----+-----+											
		ADD				EVALUATE					
*		ANALYZE				EVALUATE					
		COMPARE									
		CHECK				VERIFY,VALIDATE					
...						
+-----+-----+-----+-----+-----+-----+											
Select one or more facet terms.											
+-----+-----+-----+-----+-----+-----+											

Figure 13. Example of Facet Terms for Selected Facet Screen

Valid actions on the screen are

Exit	returns the user to the previous screen
-------------	-----------------------------------------

Explain	displays a description of the current, highlighted selection choice
Help	a screen is displayed to explain the purpose of the current screen and information about what actions are valid and how they are invoked
Process	executes a repository query based on the search criteria specified by the user
Quit	returns the user to the STARS Repository Main Menu
Reset	deselects all selected facet terms on the current screen
Select	toggle selection of individual object assignments by pressing the enter key

B.2.1.5 Keyword Search Selection Screen

The purpose of this screen shall be to show the current selection criteria for the Keyword selection method and allow this selection to be changed. An example of the Search Criteria Summary (Keyword Selection Method) screen is shown in Figure 14.

```

+-----+
|               Help      Process      Quit      Specify      eXit               |
+-----+
|               STARS Repository Keyword Specification               |
+-----+
| Current keyword value : _____ |
| New value for keyword : _____ |
+-----+
| Press [Enter] with no value to abort keyword specification. |
+-----+

```

Figure 14. Example of the Keyword Selection Criteria Summary Screen

Valid actions on the screen are

Exit	returns the user to the previous screen
Help	a screen is displayed to explain the purpose of the current screen and information about what actions are valid and how they are invoked
Process	executes a repository query based on the search criteria specified by the user
Quit	returns the user to the STARS Repository Main Menu
Specify	allows assignment or reassignment of values to attributes

B.2.1.6 STARS Repository Search Criteria Summary

The purpose of this screen shall be to list the search fields associated with the component type selected. An example of the STARS Repository Search Field Summary screen is shown in Figure 15 on page 39.

	Explain	Help	Process	Quit	Reset	Specify	eXit
	STARS Repository Search Field Summary						
Sel	SEARCH FIELDS (Current Search Value)						
*	AUTHOR (Like GRADY%)						
*	CLASS (1.1 Software Building Blocks)						
*	DATE_ENTERED (Like __-DEC-1989)						
	LANGUAGE						
	KEYWORD						
...	...						
	Select/deselect one or more search criteria before search processing.						

Figure 15. Example of the Summary Screen Layout

Valid actions on the screen are

Exit	returns the user to the previous screen
Explain	displays a description of the current, highlighted selection choice
Help	a screen is displayed to explain the purpose of the current screen and information about what actions are valid and how they are invoked
Process	executes a repository query based on the search criteria specified by the user
Quit	returns the user to the STARS Repository Main Menu
Reset	resets (clears) all selected objects on the current screen
Select	toggle individual object assignments by highlighting the object and then pressing the enter key.
Specify	allows assignment or reassignment of values to attributes

B.2.2 COMPONENT PROCESSING

This section describes the windows that display information and relations about components.

B.2.2.1 Components returned from search

The purpose of this screen shall be to list the components that matched the selection criteria. An example of the Components Returned from Search screen is shown in Figure 16 on page 40.

Help	Info	Parts	Quit	Reports	Subscribe	Versions	eXit
Components returned from search							
NAME						TYPE	
CHARACTER-SET_PACKAGE						SOFTWARE	
DATE_PACKAGE						SOFTWARE	
FILE_OPERATIONS						SOFTWARE	
...						...	
Select component before processing.							

Figure 16. Example of the Components Returned From Search Screen

Valid actions on the screen are

Exit	returns the user to the previous screen
Help	a screen is displayed to explain the purpose of the current screen and information about what actions are valid and how they are invoked
Info	provides the user with information about the object highlighted by the cursor
Parts	displays the list of parts that make up the component highlighted by the cursor. Some component parts are optional and other are required for the component to be considered 'complete'. An asterisk will be displayed to the left of the required parts.
Quit	returns the user to the STARS Repository Main Menu
Reports	allows the user to access the change and problem reports associated with the object highlighted by the cursor
Subscribe	allows the user to subscribe to the object highlighted by the cursor
Versions	displays on line component version data for the object highlighted by the cursor

B.2.2.2 Versions of a Selected Component

The purpose of this screen shall be to provide version information about the selected component. An example of the Version of selected component screen is shown in Figure 17 on page 41.

Help	Info	Parts	Quit	Reports	Subscribe	eXit
Versions of SOFTWARE			: EXTENDED CHARACTER UTILITIES			
VERSION	DATE	REASON				
01.00	28-NOV-1988	Original version				
01.01	14-MAR-1989	Respond to Discrepancy #24				
...				
Select component version before processing.						

Figure 17. Example of Versions of Selected Component Screen

Valid actions on the screen are

Exit	returns the user to the previous screen
Help	a screen is displayed to explain the purpose of the current screen and information about what actions are valid and how they are invoked
Info	provides the user with information about the object highlighted by the cursor
Parts	displays the list of parts that make up the component highlighted by the cursor. Some component parts are optional and other are required for the component to be considered 'complete'.
Quit	returns the user to the STARS Repository Main Menu
Reports	allows the user to access the change and problem reports associated with the object highlighted by the cursor
Subscribe	allows the user to subscribe to the object highlighted by the cursor

B.2.2.3 Attributes of a Selected Component

The purpose of this screen shall be to show the attributes associated with the selected component. An example of the Attributes of selected component screen is shown in Figure 18 on page 42.

Help Quit eXit	
Attributes of SOFTWARE : LIST DOUBLE BOUNDED MANAGED	
ATTRIBUTE	VALUE
AUTHOR	GRADY BOOCH
BOUNDING_INFO	BOUNDED
CLASS	Lists
...	...
Press exit key to return after scrolling through information.	

Figure 18. Example of the Attributes of Selected Component Screen

Valid actions on the screen are

Exit returns the user to the previous screen
Help a screen is displayed to explain the purpose of the current screen and information about what actions are valid and how they are invoked
Quit returns the user to the STARS Repository Main Menu

B.2.2.4 Report Operations Selection Screen

The purpose of this screen shall be to provide the ability to work with and or view problem or change reports associated with the selected component. An example of the Change Report Options screen is shown in Figure 19.

Help Quit eXit	
Change Report Options	
Work with Problem Report/Change Request	
View Problem Reports	
View Change Reports	
Select one of the options and press [Enter].	

Figure 19. Example of Change Report Options Screen

Valid actions on the screen are

Exit returns the user to the previous screen
Help a screen is displayed to explain the purpose of the current screen and information about what actions are valid and how they are invoked
Quit returns the user to the STARS Repository Main Menu

Select allows the user to select the object highlighted by the cursor. If that object is the 'View Problem Reports' item, a screen similar to the one described in "Problem Reports for Selected Component" on page 43 will be displayed. If that object is the 'View Change Reports' item, a screen similar to the one described in "Component Changes to Selected Component Version" will be displayed.

B.2.2.5 Problem Reports for Selected Component

The purpose of this screen shall be to allow the user to review problem reports for the selected component type. An example of the Problem Reports for "selected component type" screen is shown in Figure 20.

-----+-----	
Browse	Help Info Quit eXit
-----+-----	
Problem Reports for CALENDAR UTILITY	
-----+-----	
Title	Date
-----+-----	
Unable to convert to European standard	06-DEC-88
Calendar utility test program report	03-NOV-88
...	...
-----+-----	
Select one of the reports and press [Enter] to Browse.	
-----+-----	

Figure 20. Example of the Selected Component Problem Report Screen

Valid actions on the screen are

Browse	displays the highlighted problem report in the mode selected from the previous screen
Exit	returns the user to the previous screen
Help	a screen is displayed to explain the purpose of the current screen and information about what actions are valid and how they are invoked
Info	provides the user with information about the object highlighted by the cursor
Quit	returns the user to the STARS Repository Main Menu
Select	selects the currently highlighted report for browsing

B.2.2.6 Component Changes to Selected Component Version

The purpose of this screen shall be to provide information about component changes. An example of the Component Changes to Selected Component Version screen is shown in Figure 21 on page 44.

Browse			Help	Info	Quit	eXit
Component Changes to CALENDAR UTILITY						
Version	Change_date	Title				
01.01	31-OCT-1989	Leap Year Upgrade				
01.00	04-JUL-1989	Support European Format				
Select one of the reports and press [Enter] to Browse.						

Figure 21. Example of Component Changes to Selected Component Screen

Valid actions on the screen are

Browse views the highlighted change report
Exit returns the user to the previous screen
Help a screen is displayed to explain the purpose of the current screen and information about what actions are valid and how they are invoked
Info provides the user with information about the object highlighted by the cursor
Quit returns the user to the STARS Repository Main Menu
Select selects the highlighted change report for browsing

B.2.2.7 Problem Report Options

The purpose of this screen shall be to allow several operations upon a problem report. Once the user selects an option by moving to the appropriate line and pressing Enter, they will be placed in the corresponding window as shown below.

Submit ... Figure 23 on page 46

Answer ... Figure 24 on page 47

View ... Figure 25 on page 48

An example of the Problem Report Options screen is shown in Figure 22 on page 45.

Help	Quit	eXit
Problem Report Options		
Submit a problem report		
Answer a problem report		
View a problem report		
Change respondent to problem report		
Select one of the options and press [Enter].		

Figure 22. Example of Problem Report Options Screen

Valid actions on the screen are

Exit	returns the user to the previous screen
Help	a screen is displayed to explain the purpose of the current screen and information about what actions are valid and how they are invoked
Quit	returns the user to the STARS Repository Main Menu
Select	selects the highlighted object for operation

B.2.2.8 Component Problem Report Submission

The purpose of this screen shall be to allow the submission of a problem report. An example of the Component Problem Report Submission screen is shown in Figure 23 on page 46.

Cancel	Help	Quit	eXit
Component Problem Report Submission			
Component Name : _____			
Component Type : _____		Component Version : _____	
Component ID number : _____			
Your username : _____			
Problem Title: _____			
Problem Description:			

...			
Use [Tab] to move between sections, arrow keys to scroll text.			

Figure 23. Example of Problem Report Submission Screen

Valid actions on the screen are

- Cancel** returns the user to the previous screen without submitting the problem report
- Exit** returns the user to the previous screen
- Help** a screen is displayed to explain the purpose of the current screen and information about what actions are valid and how they are invoked
- Quit** returns the user to the STARS Repository Main Menu

B.2.2.9 Component Problem Report Submission

The purpose of this screen shall be to allow the answering of a problem report. An example of the Component Problem Report Answer screen is shown in Figure 23.

Cancel	Help	Quit	eXit
Component Problem Report Answer			
Problem Title: _____			
Problem report number : _____			
Your username : _____			
Answer Description:			

...			
Use [Tab] to move between sections, arrow keys to scroll text.			

Figure 24. Example of Problem Report Answer Screen

Valid actions on the screen are

- | | |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------|
| Cancel | returns the user to the previous screen without submitting the problem report |
| Exit | returns the user to the previous screen |
| Help | a screen is displayed to explain the purpose of the current screen and information about what actions are valid and how they are invoked |
| Quit | returns the user to the STARS Repository Main Menu |

B.2.2.10 Component Problem Report Viewing

The purpose of this screen shall be to allow the viewing of a problem report. An example of the Component Problem Report Viewing screen is shown in Figure 25 on page 48.

Help	Quit	eXit
Component Problem Report Viewing		
Problem report number : _____		
Problem Title: _____		
Problem Description:	Reported by : _____	

...		
Answer Description:	Answered by : _____	

...		
Use [Tab] to move between sections, arrow keys to scroll text.		

Figure 25. Example of Problem Report Answer Screen

Valid actions on the screen are

- | | |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------|
| Exit | returns the user to the previous screen |
| Help | a screen is displayed to explain the purpose of the current screen and information about what actions are valid and how they are invoked |
| Quit | returns the user to the STARS Repository Main Menu |

B.2.2.11 User Subscription Options

The purpose of this screen shall be to provide the user with a facility for subscribing to components. An example of the User Subscription Options screen is shown in Figure 26 on page 49.

Help	Quit	Select	eXit
User Subscriptions Options			
Copy a Component			
Subscribe to a Component			
Subscribe to and Copy Component			
Display All Subscriptions			
Select from the options and press [Enter].			

Figure 26. Example of User Subscriptions Options Screen

Valid actions on the screen are

Exit returns the user to the previous screen
Help a screen is displayed to explain the purpose of the current screen and information about what actions are valid and how they are invoked
Quit returns the user to the STARS Repository Main Menu
Select allows the user to perform the object highlighted by the cursor.

If the object is 'Copy a Component', a screen similar to Figure 27 is displayed. If the object is 'Display All ...', a screen similar to Figure 28 on page 50 shall be provided.

Browse				Copy		Help		Info		Quit		Reset		eXit	
Parts of CALENDAR UTILITY															
Sel		Part Type				Part Name						Version			
*		BODY				CALENDAR UTILITY BODY						01.00			
*		DESCRIPTION				CALENDAR UTILITY DESCRIPTION						01.00			
		HISTORY				CALENDAR UTILITY HISTORY						01.00			
*		SPEC				CALENDAR UTILITY SPEC						01.00			
...				
Select/deselect Component Parts to be copied															

Figure 27. Example of Component Parts for Component Selected for Copy

Valid actions on the screen are

Browse views the textual information associated with the highlighted part
Copy copies the textual information associated with the highlighted part to an external file
Exit returns the user to the previous screen
Help a screen is displayed to explain the purpose of the current screen and information about what actions are valid and how they are invoked

Info	provides the user with information about the object highlighted by the cursor
Quit	returns the user to the STARS Repository Main Menu
Reset	resets (deselects) all selected parts
Select	toggles selection of part for inclusion in copy process

B.2.2.12 Subscribed to Items

The purpose of this screen shall be to provide the user with a method of tracking component subscriptions. An example of the Subscribed to Items screen is shown in Figure 28.

Drop Help Info Quit Reset eXit		
Subscribed to Items		
Sel	Component Name	(Type)
*	CALENDAR UTILITY	(SOFTWARE)
*	EXTENDED CHARACTER UTILITIES	(SOFTWARE)
	IBM CDRL 1590	(DOCUMENT)
	STACK PACKAGE	(SOFTWARE)
*	WINDOW MANAGER	(SOFTWARE)
...	...	
Select/deselect one or more subscriptions to drop.		

Figure 28. Example of Subscribed to Items Screen

Valid actions on the screen are

Drop	allows the user to drop their subscription to the selected components
Exit	returns the user to the previous screen
Help	a screen is displayed to explain the purpose of the current screen and information about what actions are valid and how they are invoked
Info	provides the user with information about the object highlighted by the cursor
Quit	returns the user to the STARS Repository Main Menu
Reset	resets (clears) all selections indicated on the current screen
Select	toggles selection of subscription for drop process

B.2.3 PART PROCESSING

This section describes the windows that display information about component parts and there relations.

B.2.3.1 Parts of Selected Component

The purpose of this screen shall be to show the component parts associated with the selected component. An example of the Parts of Selected Component screen is shown in Figure 29 on page 51.

+-----+-----+-----+-----+-----+-----+-----+-----+-----+															
Browse		Changes		cOpy		Help		Info		Quit		Versions		eXit	
-----+-----+-----+-----+-----+-----+-----+-----+-----															
Parts of SOFTWARE				: ADA PROGRAM FLOW ANALYSIS TOOL				(V01.00)							
-----+-----+-----+-----+-----+-----+-----+-----+-----															
TYPE				PART				VERSION							
-----+-----+-----+-----+-----+-----+-----+-----+-----															
BODY				APFAT MAIN DRIVER				01.00							
COMPONENT				USER INTERFACE PACKAGE				01.02							
DESIGN				APFAT DESIGN NOTES				01.00							
RIGHTS				STANDARD GENERAL PUBLIC RELEASE				01.03							
SPEC				APFAT MAIN DRIVER				01.00							
...										
-----+-----+-----+-----+-----+-----+-----+-----+-----															
Select a Component Part before processing.															
-----+-----+-----+-----+-----+-----+-----+-----+-----															

Figure 29. Example of Parts of a Selected Component Screen

Valid actions on the screen are

Browse	displays the part highlighted by the cursor in browse mode
Changes	displays a list of all change reports associated with this part
Copy	initiates the copy process for the user selected component part
Exit	returns the user to the previous screen
Help	a screen is displayed to explain the purpose of the current screen and information about what actions are valid and how they are invoked
Info	displays attribute information associated with the highlighted part
Quit	returns the user to the STARS Repository Main Menu
Versions	displays a line component version data for the object highlighted by the cursor

B.2.3.2 Version of selected part

The purpose of this screen shall be to provide version information about the selected part. An example of the Version of selected part screen is shown in Figure 30 on page 52.

Browse	Changes	cOpy	Help	Info	Quit	Using	eXit
Versions of RIGHTS				: STANDARD GENERAL PUBLIC RELEASE			
VERSION	DATE	REASON					
01.03	18-DEC-1989	Update to meet new regulations					
01.02	04-AUG-1989	Response to problem #42					
01.01	12-MAR-1989	Include non-government rights					
01.00	28-NOV-1988	Original version					
Select a Component Part Version before processing.							

Figure 30. Example of Versions of Selected Component Part Screen

Valid actions on the screen are

- Browse** view the selected component part in browse mode
- Changes** list all change reports associated with the selected part version
- Copy** copies the selected component part into a file specified by the user via the screen shown in Figure 33 on page 54.
- Exit** returns the user to the previous screen
- Help** a screen is displayed to explain the purpose of the current screen and information about what actions are valid and how they are invoked
- Info** provides the user with information about the object highlighted by the cursor
- Quit** returns the user to the STARS Repository Main Menu
- Using** displays the components and their versions that use the selected part. This information is displayed in Figure 34 on page 55.

B.2.3.3 Information for Selected Part

The purpose of this screen shall be to show the attributes associated with the selected part. An example of the Attributes of Selected Part screen is shown in Figure 31 on page 53.

Help	Quit	eXit
Information for Part		
Name : CALENDAR UTILITY		
Type : BODY		
Default Version : 01.01		
Owner : DOE, JOHN		
Date Entered : 12-NOV-1988		
Press the exit key to return.		

Figure 31. Example of the Attributes of Selected Part Screen

Valid actions on the screen are

Exit returns the user to the previous screen
Help a screen is displayed to explain the purpose of the current screen and information about what actions are valid and how they are invoked
Quit returns the user to the STARS Repository Main Menu

B.2.3.4 Changes to selected Part Version

The purpose of this screen shall be to provide information about part changes. An example of the Changes to a component part version screen is shown in Figure 32.

Browse			Help			Info			Quit			eXit		
Part Changes to CALENDAR UTILITY (BODY)														
Version		Change_date			Title									
01.01		31-OCT-1989			Leap Year Upgrade									
01.00		04-JUL-1989			Support European Format									
...										
Select a report and press [Enter] to Browse														

Figure 32. Example of Component Changes to Selected Part Screen

Valid actions on the screen are

Browse views the highlighted change report
Exit returns the user to the previous screen
Help a screen is displayed to explain the purpose of the current screen and information about what actions are valid and how they are invoked

Info	provides the user with information about the object highlighted by the cursor
Quit	returns the user to the STARS Repository Main Menu

B.2.3.5 Component Part Copy

The purpose of this screen shall be to allow the user to specify a file to copy a single part to. An example of the Component Part Copy screen is shown in Figure 33.

```

+-----+
|               Help      Quit      eXit               |
+-----+
|               Copy Part Version to External File       |
+-----+
| Part Version : WINDOW_MANAGER (BODY) V01.00           |
| External file name: _____                       |
+-----+
| Press [Enter] with no file name to cancel operation.  |
+-----+

```

Figure 33. Example of Component Part Copy Screen

Valid actions on the screen are

Exit	returns the user to the previous screen
Help	a screen is displayed to explain the purpose of the current screen and information about what actions are valid and how they are invoked
Quit	returns the user to the STARS Repository Main Menu

B.2.3.6 Components using selected component part

The purpose of this screen shall be to provide information about which components in the repository are using the selected part. An example of the Components Using Selected Component Parts screen is shown in Figure 34 on page 55.

Help	Info	Parts	Quit	Reports	Subscribe	Versions	eXit
Comps using RIGHTS : TREE ARBITRARY DOUBLE BOUNDED MANAGED (V01.00)							
NAME					TYPE		VERSION
LIST DOUBLE BOUNDED MANAGED					SOFTWARE		01.00
LIST DOUBLE BOUNDED CONTROLLED					SOFTWARE		01.00
LIST DOUBLE BOUNDED CONTROLLED					SOFTWARE		01.01
LIST DOUBLE BOUNDED UNMANAGED					SOFTWARE		01.00
LIST DOUBLE BOUNDED UNMANAGED					SOFTWARE		01.01
LIST DOUBLE BOUNDED UNCONTROLLED					SOFTWARE		01.00
...				
Select a component before processing.							

Figure 34. Example of Components Using a Part Version Screen

Valid actions on the screen are

Exit	returns the user to the previous screen
Help	a screen is displayed to explain the purpose of the current screen and information about what actions are valid and how they are invoked
Info	provides the user with information about the object highlighted by the cursor
Parts	displays the component parts composing the selected component
Quit	returns the user to the STARS Repository Main Menu
Reports	Displays the problem and change report options for the selected component
Subscribe	displays the subscription/copy options
Versions	displays on line component version data for the object highlighted by the cursor

B.3 REPOSITORY SUPPORT OVERVIEW

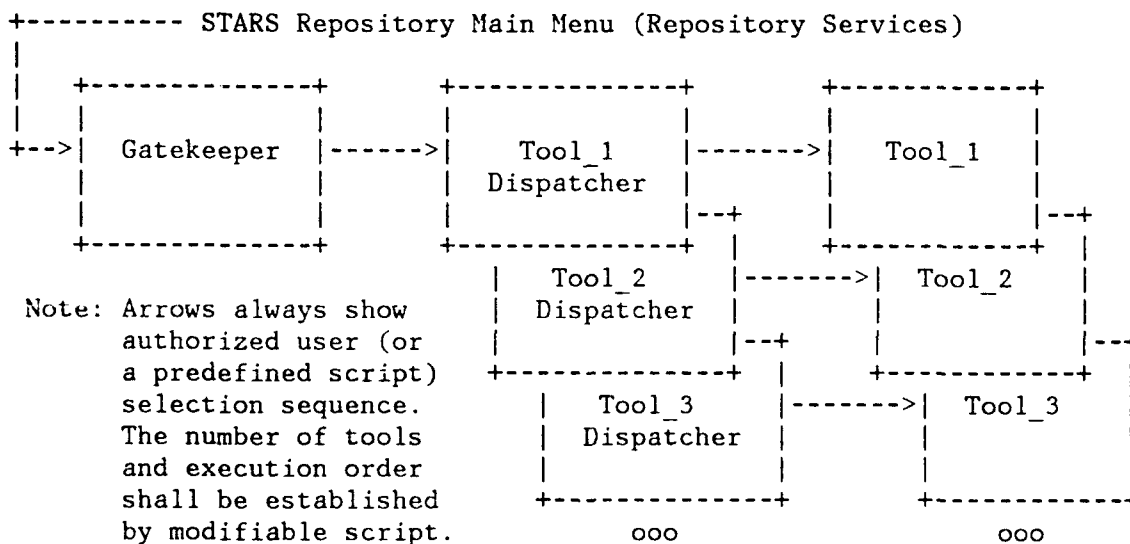
The search and access process discussed in the preceding paragraphs assumed a specific set of processing support is available, i.e., to ensure a useful set of components is available in the Repository and that accurate information is available that describe the authorized users. This information is used to provide default values, e.g., in the report generation process.

B.3.1 COMPONENT SUPPLY AND FILTERING

The component supply process will evolve as the repository matures and the scope of the contents expands. The exact processing and sequence of processing shall be as defined in the STARS Repository Operation and Procedures (e.g., CDRL 1470) and in the Repository Guidebook (e.g., CDRL

1550) documents. These documents define the processing (either manual or automated) that are required to add a component to the repository or to remove a component from the repository.

Figure 35 includes a dispatcher for each tool which may be required if the tool is not compatible with the Repository System, e.g., the Virtual Operating System Interface (VOSI) environment may require preprocessing or post-processing of input or output data. The control process is referred to as the Gatekeeper and the unique processing required to interface individual tools with the Repository System is referred to as the Dispatcher. The Gatekeeper shall interface with the STARS Repository to assist the Repository Manager or Librarian.



B.3.2 STARS REPOSITORY REGISTRATION FORMS

The following forms are for information that will usually be obtained from a prologue section of a component submitted to the repository. It may not be feasible, however, for some types of components to have this prologue. It is also likely that information being transferred from some other repository en masse will not have this prologue. For these reasons the forms are provided.

B.3.2.1 Component Registration Form

The first form, Figure 36 on page 58, contains data that will vary depending upon the type of component being submitted. It should be remembered that this is just an sample form.

STARS REPOSITORY SYSTEM
Component Registration Form

Name: _____

Version: _____ Old Component ID: _____

Access Type: _____ Contract ID: _____

Author ID: _____ Owner ID: _____

Attributes: (Required unless supplied in the Component Prologue)

Type: SOFTWARE Class: _____

Media: _____ Size: _____

Structure: _____ Format: _____

Concurrency: _____ Bounded: _____

Media Mgmt.: _____ Iteration: _____

Contact: _____

User Identification Name: _____

Organization Identification Number: _____

----- Completed by Repository System -----

Repository Entrance Date: _____

New Component Identification Number: _____

Notes: Attach Certificate of Origin Forms to all new components
and revised/duplicate forms to new versions of components.

Attach Change Reports for all new versions of registered
components/component parts (by Name and Version Number).

Figure 36. Example of Component Registration Form

B.3.2.2 User Registration Form

This form is used to enter in data for any person, regardless of their role in the Repository (Supplier, Manager, Re-user, Owner, etc.). This information will be entered only once for each person and will be modifiable.

STARS REPOSITORY SYSTEM User Registration Form

Name: _____
 (First) (MI) (Last)

Title: _____ Position: _____

Nationality: _____ U.S.Status _____

Organization Identification Number: _____

Requested User Identification Name: _____

Optional Network Address: _____

----- Completed by Repository System -----

Requested User Identification Name: _____

Figure 37. Example of User Registration Form

B.3.2.3 Organization Registration Form

This form is used to enter in data for any organization represented in the Repository. This information will be entered only once for each person and will be modifiable.

STARS REPOSITORY SYSTEM
Organization Registration Form

Organization: _____

Address: _____

(City) (State) (ZIP-Code)

(Country) (Phone Number)

----- Completed by Repository System -----

Organization Identification Number: _____

Qualified Contractor Access Number: _____

Note: The above registration numbers may be based on numbers assigned to the attached copy of DD FORM 2245 approved by:

United States/Canada Joint Certification Office
Defense Logistics Service Center, Federal Center
Battle Creek, MI 49017-3084

Attach one or more completed User Registration forms and indicate the primary (and optional alternates) coordinators.

Figure 38. Example of Organization Registration Form

B.3.2.4 Contract Registration Form

This form is used to enter in data for any contract represented in the Repository. This information will be entered only once for each person and will be modifiable.

STARS REPOSITORY SYSTEM
Contract Registration Form

Contract Number: _____

Contract Name: _____

Contracting Agency: _____

Contact Name: _____

Address: _____

Optional Network Address: _____

----- Completed by Repository System -----

Organization Identification Number: _____

Contract Contact Identification Name: _____

Figure 39. Example of Contract Registration Form

APPENDIX C. REPOSITORY DETAIL DESIGN DOCUMENT

C.1 DATA DICTIONARY

Data Item	Format	Definition
Component id	Integer	Unique number identifying a particular component of a particular type.
Component name	Char (40)	Name of a reusable object
Component type	Char (15)	Type of the reusable object. Examples are software, documents, plans.
Component version id	Integer	Unique number identifying a particular version of a particular component of a particular type.
Part id	Integer	Unique number identifying a particular part of a particular type.
Part name	Char (40)	Name of a text file which contains specific information about a component.
Part type	Char (15)	Type of the information to be used for a component. Examples are abstract, code, test driver, license.
Part version id	Integer	Unique number identifying a particular version of a particular part of a particular type.
Person id	Integer	Unique number identifying a specific person who assumes one or several roles within the repository.
Owner	Person id	Name of the individual who owns and is responsible for a component or part. Note that the same owner is responsible for all versions of the component part.
Contributor	Person id	The individual who contributed the component to the repository.
Contact	Person id	The individual within an organization who is the representative for that organization in regard to one aspect of the component, such as the contract.

Data Item	Format	Definition
Version	Char (6)	Used to differentiate objects which are the same except for enhancements and corrections. Also appears as: component_version, default_version, and part_version.
Version reason	Char (80)	Used to document the reason that a new version of a component or part has been entered into the system.
Date entered	DD-MMM-YYYY	Denotes the date that the associated object was entered into the system.
Part required	Char (1)	Denotes whether a specific part type associated with a specific component type is required for a complete component definition or operation.
Required Indicator	Char (3)	Denotes whether the object is Required (REQ), Optional (OPT), Multiple values allowed (MUL), or Multiple values/Optional (MOP).
Search name	Char (20)	The name of a search field which can be used to locate components. Also appears as: Facet_name and Attribute_name.
Search type	Char (13)	Indicates the type of search field: Facet, Alphanumeric attribute, Numeric attribute, or Keyword.
Facet term	Char (20)	A single entry in the enumerated list of values for a facet.
Attribute length	Integer	The maximum length allowed for the values of a particular alphanumeric attribute.
Upper bound	Integer	The maximum value that can be specified for a numeric attribute.
Lower bound	Integer	The minimum value that can be specified for a numeric attribute.
Description	Char (80)	Provides some additional information about the object with which it is associated.
Class depth	Integer	Indicates the depth in the hierarchy of the associated class.

Data Item	Format	Definition
Class number	Char (40)	Provides a numeric designation in character form for the associated class. An example would be "1.1.3.2".
Level number	Integer	Actually is 8 fields: Lev1, Lev2, ..., Lev8. Indicates the number of the class at each level defined for the class.
Class seq	Integer	Random number (actually assigned as current highest +1) that is used to associate class entries with components.
Report	Integer	Unique number identifying an external file that contains textual information connected with a problem, change, component part, or other object.
Report id	Integer	Unique number identifying a particular report.
Organization id	Integer	Unique number identifying a particular organization represented in the repository.
Nation id	Char (3)	Unique character string identifying a national entity represented in the repository.
US Status	Char (1)	Values: Y (the object is considered to be an entity under USA laws and regulations), or N (the object is not considered as such and may be under certain regulations as to shipping, passing of restricted information, etc.)
Export Control Status	Char (80)	String containing any regulations pertaining to export of information in the repository.

Table 1. Repository Data Dictionary: This table contains the names, formats and definition of all fields used in the Repository database.

C.2 DATA NORMALIZATION

The abstracted data objects for the STARS Repository will be developed / presented in this section. The objects are divided into the following groups for purposes of discussion:

1. Component Objects
2. Search Objects
3. Component Construction Objects
4. Environment Support / Information Objects

Each of these groups will have the following information included:

Definitions	Defines the objects
Relationships	Gives a pictorial representation of the relationships among the objects in the group
Content	Lists the fields which make up the object. Also identifies keys.

C.2.1 COMPONENT OBJECTS

Component objects define the component units and the relations among the text units.

C.2.1.1 Definitions

Component A component is the collection of information which completely describes a reusable object. The Repository will support multiple versions of a component.

Part A part is a single instance of information which can be used to describe a component. The Repository will support multiple versions of a part.

Component Parts Each component version has a collection of parts (with specific versions) specified. Each component version has at most one part of a particular type, unless a part is another component, in which case there may be multiple parts of this type.

C.2.1.2 Relationships

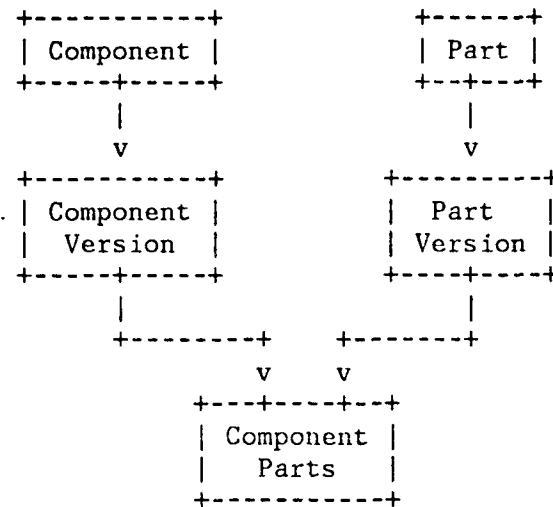


Figure 40. Component Object Relationships

C.2.1.3 Object Content

Relation	Key	Fields
Component	<ul style="list-style-type: none"> Component id 	<ul style="list-style-type: none"> Component name Component type Default version Owner Date entered
	Used to record the existence of a component. Search fields describe objects at this level.	
Component Version	<ul style="list-style-type: none"> Component version id 	<ul style="list-style-type: none"> Component id of parent component Component version Reason for version Date entered
	Used to manage multiple component versions.	

Relation	Key	Fields
Part	<ul style="list-style-type: none"> Part id 	<ul style="list-style-type: none"> Part name Part type Default version Owner Date entered
	A unit of text to be managed and controlled by the STARS repository. This element may be an abstract, license agreement, code, test case, etc. and may be referenced by more than one component.	
Part Version	<ul style="list-style-type: none"> Part version id 	<ul style="list-style-type: none"> Part id for parent part Reason for version Date entered
	Used to manage multiple part versions.	
Component Part	<ul style="list-style-type: none"> Component type Part type 	<ul style="list-style-type: none"> Part id
	Each component has a list of part types which can be specified. Only one part can be defined for each part type.	

Table 2. Component Object Content: The key and non-key fields are shown for each component object. A description of each object is also provided.

C.2.2 SEARCH OBJECTS

Search objects describe the kinds of fields which can be used to classify and select components.

C.2.2.1 Definitions

Search Fields A search field describes some property of a component.

Hierarchy A tree structured classification scheme.

Hierarchy Entries The individual nodes in a classification tree. Each node can have child nodes or matching components.

Facets A search field which has a predefined, enumerated list of valid character values and describes one particular trait.

Facet Terms The enumerated list of valid character values for a specific Facet.

Character Attributes A search field which does not have a predefined list of valid values and is composed of character values.

Numeric Attributes A search field which does not have a predefined list of valid values and is composed of numeric values.

C.2.2.2 Relationships

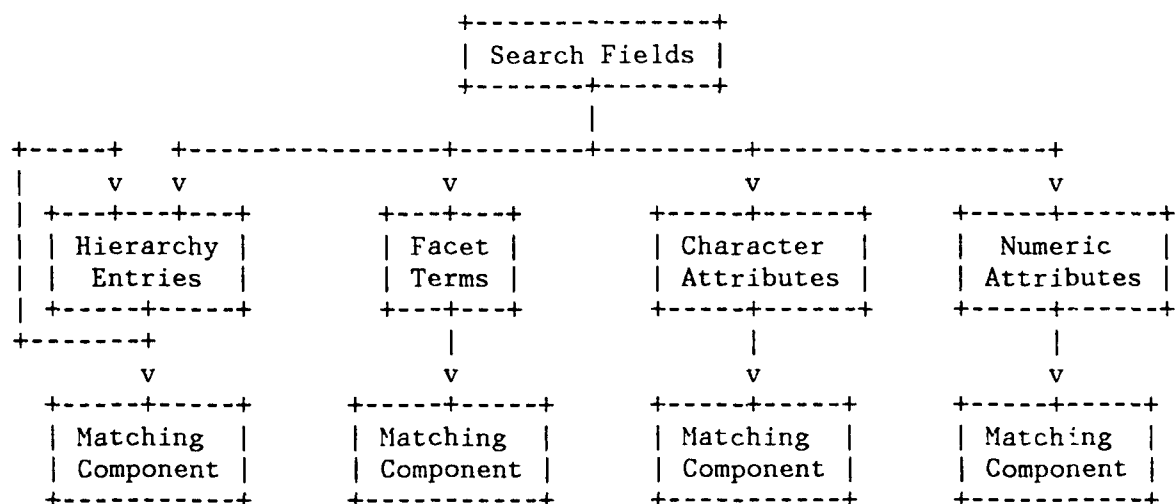


Figure 41. Search Object Relationships

C.2.2.3 Object Content

Relation	Key	Fields
Search fields	<ul style="list-style-type: none"> • Search name 	<ul style="list-style-type: none"> • Search type • Description
	Defines a component classification view (i.e. Language, Scope, etc.). Search type is hierarchy, facet, character attribute, numeric attribute, or keyword.	
Hierarchy entries	<ul style="list-style-type: none"> • Lev1 • Lev2 • . . . • Lev8 	<ul style="list-style-type: none"> • Class name • Class depth • Class number • Class seq • Class description
	Each hierarchical entry contains information to allow ordered traversal of the hierarchy and provides the capability to query on any class.	
Facet terms	<ul style="list-style-type: none"> • Facet value key <ul style="list-style-type: none"> - Facet name - Facet term 	<ul style="list-style-type: none"> • Description • Aliases
	Each facet contains an enumerated list of facet values which may be used during component supply and retrieval. Nothing is implied by the ordering of the terms within a facet.	
Components with Facet Terms	<ul style="list-style-type: none"> • Facet value key (name+term) • Component id 	
	The components which have a particular character facet value.	
Numeric Attribute	<ul style="list-style-type: none"> • Attribute Name 	<ul style="list-style-type: none"> • Upper bound • Lower Bound
	Numeric attributes require the definition of valid ranges.	
Components with Numeric Attributes	<ul style="list-style-type: none"> • Attribute value key (name+value) • Component id 	
	The components which have a particular numeric attribute value.	
Character Attribute	<ul style="list-style-type: none"> • Attribute Name 	<ul style="list-style-type: none"> • Attribute length
	Character attributes require the definition of valid ranges.	

Relation	Key	Fields
Components with Character Attributes	<ul style="list-style-type: none"> Attribute value key (name+value) Component id 	
	The components which have a particular numeric character value.	

Table 3. Search Object Content: The key and non-key fields are shown for each search object. A description of each object is also provided.

C.2.3 COMPONENT CONSTRUCTION OBJECTS

Component construction objects dictate what kind of information and search mechanisms are available for different types of components

C.2.3.1 Definitions

Component Type

A component type dictates which parts and search criteria are valid or required for a component of the type.

Part Type Each part has a valid type which determines what roles that part type may fill for a component.

Component Part Contents

Each component type defines a collection of part types that are valid or required.

Component Search Contents

Each component type defines a collection of search fields that are valid or required.

C.2.3.2 Relationships

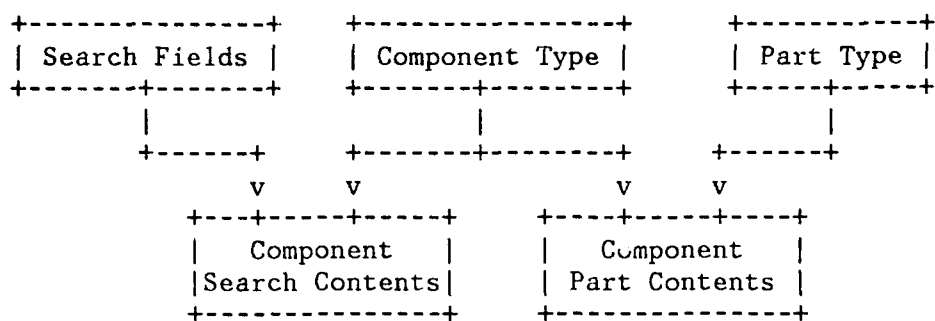


Figure 42. Component Content Relationships

C.2.3.3 Object Content

Relation	Key	Fields
Component Content Parts	<ul style="list-style-type: none"> • Component type • Part type 	<ul style="list-style-type: none"> • Required indicator
	A "template" which dictates which parts constitute a component.	
Component Content Search Fields	<ul style="list-style-type: none"> • Component type • Search field name 	<ul style="list-style-type: none"> • Required indicator
	A "template" which dictates which search fields are required for a component.	

Table 4. Component Content Objects: The key and non-key fields are shown for each component content object. A description of each object is also provided.

C.2.4 ENVIRONMENT SUPPORT / INFORMATION OBJECTS

Environment Support / Information Objects keep track of information needed to support the development system.

C.2.4.1 Definitions

Problem report

A report submitted by any valid user stating a problem found with a particular version of a component.

Change report

A report submitted by the owner of an object stating the changes made to the object resulting in a new version of the object.

C.2.4.2 Relationships

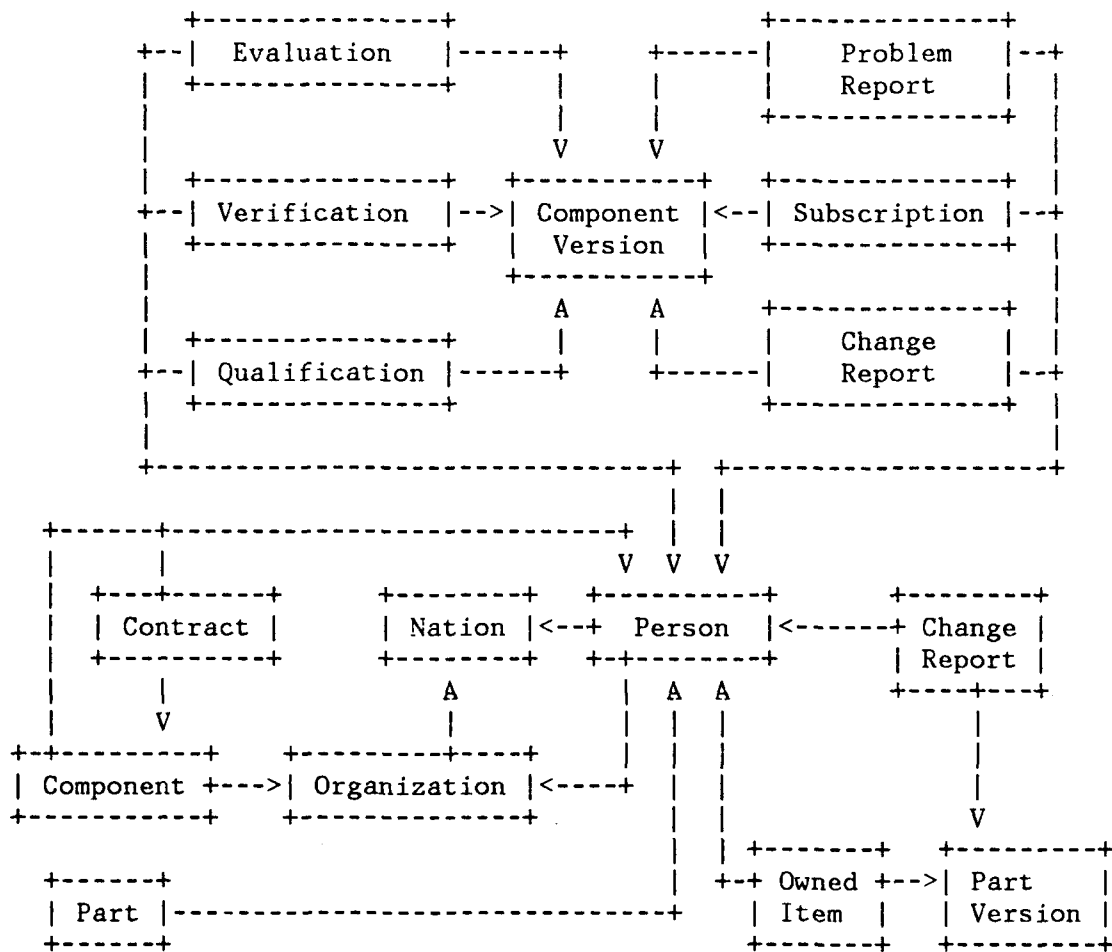


Figure 43. Environment Support / Information Object Relationships

C.2.4.3 Object Content

Relation	Key	Fields
Evaluation	<ul style="list-style-type: none"> • Component version id • Evaluator 	<ul style="list-style-type: none"> • Evaluation date • Rating • Report
	An evaluation of a version of a component and the determined rating based on the acceptance criteria.	
Verification	<ul style="list-style-type: none"> • Component version id • Verifier 	<ul style="list-style-type: none"> • Verification date • Report
	A verification of the functionality and reusable quality of a component.	
Qualification	<ul style="list-style-type: none"> • Component version id 	<ul style="list-style-type: none"> • Compiled • Documented • Tested
	A qualification of the component for entry into a higher level repository library.	
Problem Report	<ul style="list-style-type: none"> • Component version id • Report id 	<ul style="list-style-type: none"> • Title • Report date • Reporter • Respondent • Resolution date • Report
	A report submitted by any user noting a problem found with a specific component version.	
Subscription	<ul style="list-style-type: none"> • Person id • Component id 	<ul style="list-style-type: none"> • Component version id • Subscription date
	A request by a user to be notified of all modifications/changes affecting a particular component.	
Change report	<ul style="list-style-type: none"> • Component version id OR • Part version id • Report id 	<ul style="list-style-type: none"> • Title • Changer • Change date • Report
	A report submitted when a version of a part or component has been changed, explaining the change(s) made.	

Relation	Key	Fields
Person	<ul style="list-style-type: none"> Person id 	<ul style="list-style-type: none"> name Network address Username Nationality
	A record established of every user, regardless of position (e.g., Qualifier, Verifier, Evaluator, Owner, Contributor, etc.)	
Organization	<ul style="list-style-type: none"> Organization id 	<ul style="list-style-type: none"> Name Address Phone US Status
	A record of each organization represented in the repository in some capacity.	
Nation	<ul style="list-style-type: none"> Nation id 	<ul style="list-style-type: none"> Name Export control status
	A record established for each nation represented in the repository.	
Contract	<ul style="list-style-type: none"> Contract id 	<ul style="list-style-type: none"> Contract number Contract name Contracting agency Contacting agency's contact
	The contract a component was developed under.	

Table 5. Environment Support / Information Objects: The key and non-key fields are shown for each component content object. A description of each object is also provided.

C.3 RELATIONAL DESIGN

This section will present processing model of data. This will be the program "view" of the data and will not change regardless of the DBMS used in implementation.

C.3.1 PROCESSING REQUIREMENTS

The component search and access processing requirements will be investigated.

In each of the processes sections, the tables will contain a dashed line ("---"). Processes above this line are table lookups, and the costs involved will not be discussed since they are assumed to be minimized.

Processes below the dashed line involve table joins or some other complex searching and will be discussed individually.

C.3.1.1 Search processes

1. List all selectors.
2. List hierarchy entries.
3. List all terms for a facet selector.
4. List valid range for a numeric selector.
5. List valid range for a character selector.
-
6. Selection of components by search values.
7. Display all search field values for a selected component.
8. Find all common search fields for the set of selected components.

Selection of components by search values: Third normal form (TNF) dictates that the components which match search values be stored by search value (object key would be search field + search value + component id). Searching on *n* fields would require that *n* table joins be performed.

Display all search field values: With the search field tables set up as previously described, an "open join" would have to be performed on *n* tables to get the values. Since the open join operation is not a standard SQL capability, the processing (and coding) for this query are even more difficult.

Find all common search fields: For the set of component selected, find all search fields which are common to all the components in the set. This capability allows the user to select additional search fields to specify in the search query.

C.3.1.2 Component processes

1. List all versions of a component.
2. List the parts used by a component version.
3. List all versions of a part.
-
4. List all components (or versions) which use a part (or versions).

List all components which use a part: In the case where all component versions are to be listed for a particular part version, this is a simple table lookup with multiple matches. In the other cases:

- All components for a particular part version
- All components for a particular part
- All component versions for a particular part

the table must be GROUPED BY and ORDERED BY the appropriate fields.

C.3.1.3 Content processes

1. List required parts for a component type.
 2. List required selectors for a component type.
-

C.3.1.4 Support processes

1. List all problem reports for a component or part
 2. List all critiques (evaluations, verifications, etc.) for a particular component
 3. List all subscribers to a component (or version)
 4. List all changes to a part version or component version
-

5. List all changes to a component or part

List all changes to a component or part: In the case where all change reports are to be listed for a particular part or component, this set of changes to versions of the component must be retrieved, indicating a join of the Component, Component Versions, and Changes tables.

C.3.2 TRANSFORMATION RATIONAL

Due to processing needs, it may be necessary to violate TNF supplied in the section above. If so, rational / justification will be presented.

The following objects will be folded into the Component object:

- Matching Component by Hierarchical Classification.
- Matching Component by Facet and Facet Term
- Matching Component by Character Attribute
- Matching Component by Numeric Attribute

This is done for efficiency of the search operation. To allow for multiple types of components, all attributes that are dependent upon the component type will be moved to a separate table reserved for these attributes. Thus, there will exist a table for each type of component defined whose columns will be the attributes unique to the particular component type. Another table will map the type-specific attributes tables to the component types. The alternative of this method would be to keep ALL attribute values in the COMPONENT table, and only have values assigned to those attributes which apply. This, however, would leave a large amount of database space unused in any system that supports more than one type of component. The facets will be considered a unique type of attribute and as such, the facet terms for a component will not be

stored in the above mentioned type-specific tables, but will be kept in a separate table.

The results of these actions will be that, at most, eight tables need to be joined: the Component table, the type-to-specific-fields table, the specific-fields table, the facet table, the hierarchy table, the person table, the organization table, and the contract table. This case occurs when at least one of each attribute type is used for the search. If one type of attribute is not used, however, then its tables will not need to be included in the search, so that the best case scenario would be a search on only those fields in the Component table. It may prove valuable to duplicate the person's name, organization's name, and contract number as character fields in the component table to prevent the join of the last three tables mentioned. This can be determined after some real-world evaluation which would determine exactly how often these tables are actually involved in the search process.

The hierarchy will be represented as a flat table with an numerical entry for each level of the classification. This is done because SQL databases do not support recursive relations. Since ordering must be maintained and duplication of class names is allowed, the primary key of this relation will be the level numbers. Joins to the component table will be performed on a meaningless number present in both tables. For example, if the user requested that all components in class "2.1.3 Data Structures" should be found, the query to perform this would be:

```
select COMPONENT
from HIERARCHY H, COMPONENT C
where H.LEV1 = 2 and H.LEV2 = 1 and H.LEV3 = 3 and H.SEQ = C.HIER_SEQ
```

This would retrieve all components whose first three levels of hierarchical classification was "2.1.3".

C.3.3 PROCESS TABLE CONTENT

Following is a listing of the relational tables to be used in processing. Each will include the following items:

- Object keys which enforce uniqueness
- Recommendations for data fields

The primary key of every table will be implemented as a unique index to enforce uniqueness. The fields which make up the unique key will be in the order listed in the table.

SQL Table	Object Key	Data Fields
Component	<ul style="list-style-type: none"> • Component id 	<ul style="list-style-type: none"> • Component_name • Component_type • Default_version • Owner • Contributor • Date_entered • Hier_Seq • (Search field...)
Component_version	<ul style="list-style-type: none"> • Component version id 	<ul style="list-style-type: none"> • Component_id • Component_version • Version_reason • Date_entered
Part	<ul style="list-style-type: none"> • Part_id 	<ul style="list-style-type: none"> • Part_name • Part_type • Default_version • Owner • Date_entered
Part_version	<ul style="list-style-type: none"> • Part_version_id 	<ul style="list-style-type: none"> • Part_id • Part_version • Version_reason • Date_entered • Report
Component_Part	<ul style="list-style-type: none"> • Component_id 	<ul style="list-style-type: none"> • Part_type • Part_version_id
Search_fields	<ul style="list-style-type: none"> • Search_name 	<ul style="list-style-type: none"> • Search_type • Description
Facet_terms	<ul style="list-style-type: none"> • Facet • Facet_term 	<ul style="list-style-type: none"> • Aliases • Description
Num_attrs	<ul style="list-style-type: none"> • Attribute_name 	<ul style="list-style-type: none"> • Upper_bound • Lower_Bound
Alpha_attrs	<ul style="list-style-type: none"> • Attribute_name 	<ul style="list-style-type: none"> • Attribute_length
Comp_keywords	<ul style="list-style-type: none"> • Keyword 	<ul style="list-style-type: none"> • Component_id
Comp_cont_parts	<ul style="list-style-type: none"> • Component_type • Part_type 	<ul style="list-style-type: none"> • Part_required
Comp_cont_Search	<ul style="list-style-type: none"> • Component_type • Search_name 	<ul style="list-style-type: none"> • Search_required
Hierarchy	<ul style="list-style-type: none"> • Lev1 • Lev2 • . . . • Lev8 	<ul style="list-style-type: none"> • Class_name • Class_number • Sequence • Class_depth

SQL Table	Object Key	Data Fields
Type_Specific_Fields	• Component_Type	• Specific Field Table Name

Table 6. Process SQL table content: This table provides the object key and data fields for each SQL table in the Repository design. Each object key is to be defined as a unique key to enforce uniqueness. Additional indexing recommendations are given in "Additional Indexes" on page 81.

In addition to these tables, another table is needed to support use the facets to describe a component. The following table will allow a component to have multiple facet terms for a particular facet.

SQL Table	Object Key	Data Fields
Component_Facet	• Facet	• Facet_term • Component_id

Table 7. Facet SQL table content: This table provides the object key and data fields for the Component_Facet table. The object key is to be defined as a unique key to enforce uniqueness. Additional indexing recommendations are given in "Additional Indexes" on page 81.

C.3.4 SUPPORT TABLE CONTENT

Following is a listing of the relational tables to be used in environment support and information. Each will include the following items:

- Object keys which enforce uniqueness
- Recommendations for data fields

The primary key of every table will be implemented as a unique index to enforce uniqueness. The fields which make up the unique key will be in the order listed in the table.

SQL Table	Object Key	Data Fields
Contract	• Contract id	• Contract_name • Contract_number • Contracting agency • Agency contact

SQL Table	Object Key	Data Fields
Evaluation	<ul style="list-style-type: none"> • Component version id 	<ul style="list-style-type: none"> • Evaluator_id • Evaluation_date • Evaluation_report • Rating
Nation	<ul style="list-style-type: none"> • Nation_id 	<ul style="list-style-type: none"> • Name • Export Control Status
Organization	<ul style="list-style-type: none"> • Organization_id 	<ul style="list-style-type: none"> • Name • Address • Phone • US_Entity
Person	<ul style="list-style-type: none"> • Person_id 	<ul style="list-style-type: none"> • Name • Employer • Network_address • Username • Nationality • US_Status
Component_Change	<ul style="list-style-type: none"> • Component version id 	<ul style="list-style-type: none"> • Title • Change_date • Changer_id • Change_Report
Part_Change	<ul style="list-style-type: none"> • Part_version_id 	<ul style="list-style-type: none"> • Title • Change_date • Changer_id • Change_Report
Problem	<ul style="list-style-type: none"> • Problem_id 	<ul style="list-style-type: none"> • Title • Report_date • Reporter_id • Resolution_date • Respondent_id • Problem_report
Qualification	<ul style="list-style-type: none"> • Component version id 	<ul style="list-style-type: none"> • Compiled • Documented • Tested • Secure
User_Subscription	<ul style="list-style-type: none"> • User_id • Component_id 	<ul style="list-style-type: none"> • Component_version • Subscription_date
Verification	<ul style="list-style-type: none"> • Component version id 	<ul style="list-style-type: none"> • Verifier_id • Verification_date • Verification_repor

Table 8. Support SQL table content: This table provides the object key and data fields for each SQL table in the Repository design that supports the environment support tools. Each ob-

ject key is to be defined as a unique key to enforce uniqueness. Additional indexing recommendations are given in "Additional Indexes" on page 81.

C.3.5 ADDITIONAL INDEXES

C.3.5.1 Component Table

An additional index will be created for each search field which is added to the Component table. Each index will consist of the search field and component_id.

C.3.5.2 Component_part Table

An additional index should be created on Part_version_id to support queries of the type "Find all components which use part...".

C.3.5.3 Hierarchy Table

An additional index should be created on Sequence so that the highest number can be found when adding a new entry.

C.3.5.4 Component_facet Table

An additional index should be created on the Component_id to support queries of the form "Find the facets that apply to component X and find their values".

C.3.5.5 Problem Table

An additional index should be created on the Component_version_id to support finding all problem reports for a specific component version.

C.3.5.6 Subscription Table

An additional index should be created on the Component_id to help find all users that are subscribed to a particular component.

C.3.5.7 Component_Change and Part_Change Tables

An additional index should be created on the Component_version_id to support finding all problem reports for a specific component version.